A distributed system for payroll queries within a large organisation

John David Bates

BSc Computing
2004/2005

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)______________________

Summary

The aim of this project is to create a distributed system to provide a structured method of submitting and managing payroll queries within a large organisation. The system will enable management from across a number of different sites to submit queries in a structured way to a centralised payroll department. Payroll personnel will be able to use the system to retrieve and manage the queries that will be stored in the database. The payroll manager will be able to query the database to produce weekly statistics for their reports. A web based solution will ensure that management with intranet access will be able to use the system without installing additional software and having to update that software when system changes are made.

Acknowledgements

I would like to thank my supervisor, Kristina Vuskovic, for the excellent report writing advice, suggestions and general encouragement throughout the project.

Thanks are owed to those people who gave up their time to be interviewed as part of the analysis phase and also those who completed the usability evaluation questionnaire. Their input to the project is valuable and much appreciated.

Thomas Anderson and Mark Brown deserve a mention for their PHP and MySQL advice over those refectory lunches.

Finally, I would like to thank the support team for their speedy response after my request for space on school's systems.

Contents

1 Introduction **1**

1.1 Problem Domain 1

1.2 Background Reading 2

1.3 Methodologies 3

1.3.1 Structured Systems and Design Methodology (SSADM) 3

1.3.2 Systems Development Life Cycle (SDLC) 4

1.3.3 Rapid Application Development (RAD) 4

1.3.4 Prototyping 5

1.3.5 Appropriate methodology 6

1.4 Main Objectives 8

1.5 Minimum Requirements 8

1.6 Additional Requirements 8

1.7 Deliverables 9

1.8 Project Schedule 9

2 Analysis **11**

2.1 Existing Solutions 12

2.2 Requirements Gathering 14

2.2.1 Interviewing 15

2.2.2 Use Cases 16

2.3 System Requirements 16

2.3.1 Functional Requirements 17

2.3.2 Non-functional Requirements 17

2.4 Proposed System 18

3 Design 19

3.1 Database 19

3.1.1 Database Management System 19

3.1.1.1 MySQL 20

3.1.1.2 PostgreSQL 20

3.1.1.3 MS SQL Server 20

3.1.1.4 MS Access 20

3.1.1.5 Appropriate DBMS 21

3.1.2 ER modelling 21

3.1.3 Deriving table design 24

3.1.4 Normalisation 25

3.2 Scripting Language 25

3.2.1 PHP 26

3.2.2 ASP and JSP 26

3.2.3 PERL 26

3.2.4 Appropriate Scripting Language 27

3.3 System Architecture 27

3.3.1 Two-tiered architecture 27

3.3.2 Three-tiered architecture 28

3.4 HCI Usability 28

4 Implementation 30

4.1 Manager 30

4.2 Payroll Personnel 31

4.3 Payroll Manager 32

4.4 Security 33

4.5 Testing 35

4.6 Instruction Manual 36

5 Evaluation 38

5.1 HCI Usability 38

5.1.1 Questionnaire 39

5.1.2 Conclusions 40

5.2 System Requirements 43

5.2.1 Satisfied Requirements 43

5.2.2 Partially-Satisfied Requirements 46

5.2.3 Un-Satisfied Requirements 46

5.3 Achievement of Project Requirements 47

5.4 Comparison to Other Solutions 47

5.5 Future Development 48

Bibliography 50

Appendix A - Personal Reflection 51

Appendix B - Instruction Manual 54

Appendix C - Payroll Query Process - Activity Diagram 55

Appendix D - Gannt Chart 57

Appendix E - Interview Notes 59

Appendix F - Individual Use Case Diagrams 63

Appendix G - Table Design - SQL statements 65

Appendix H - Usability Evaluation Questionnaire 68

Chapter 1

Introduction

This chapter forms an Introduction to the project. The problem domain is specified by the author using his own observation. The author's experience of working in the payroll department of a large organisation is extremely useful in providing a good background to the problem. The problem domain is accompanied by an activity diagram explaining the payroll query process which can be found in appendix C. A brief section covering the background research that took place in this project is also part of this chapter. The most appropaite software development methodology is selected. The project objectives along with the minimum and additional requirements are also been derived. This chapter concludes with a list of deliverables accompanied by the project schedule.

1.1 Problem Domain

The centralised payroll department of a large organisation spread across a number of sites processes the salary of thousands of employees who receive their wage on a weekly or monthly basis. If an employee feels that their wage is wrong they will report this to their manager who will then submit a query to the payroll department and await a reply.

In order for the payroll department to process the query they need a number of details relating to it, a system that ensures that all the required information to process a query exists is needed. All managers should be able to submit and retrieve replies to their queries. Also should a manager submit another query relating to one previously submitted or an employee re-queries an issue it is essential that their previous submitted queries can be quickly retrieved by the payroll department.

The payroll department aim to resolve all queries within two days of receiving them under their service level agreement therefore an efficient automated system would make sure this is met. Payroll personnel need to be able to view the history of a particular query and add comments to it to make sure they are not wasting time resolving queries already being dealt with and to hand queries over to a new person. A solution must be found that ensures queries are not lost and are resolved in order of arrival to make sure the department meets its service level agreement.

The payroll manager uses the queries to provide statistics e.g. the number of queries in the last week which were caused by the employee's manager and those that were caused by the payroll department in the form of reports to the sites which take a considerable amount of time to compile; this means that all queries must be stored for this purpose. The payroll manager would also like to monitor the performance of the payroll personnel who resolve the queries e.g. amount of queries resolved on a particular day and average time taken to resolve a query.

1.2 Background Reading

This report is based on background reading carried out in several different areas. Background reading adds structure and confidence to the processes followed in the project. Research into appropriate software development methodologies, database management systems (DBMS), scripting languages and system architectures is completed before justifying the ones choosen. The analysis phase makes use of requirements gathering techniques and UML. Whilst the design phase is based on ER modelling, normalisation and HCI usability processes. The implementation phase makes heavy use of a scripting language reference book and documentation guidelines. Finally the evaluation phase is based on researched heuristic evaluation.

1.3 Methodologies

A methodology can be defined as *”a recommended series of steps and procedures to be followed in the course of developing an information system ”* [3] which should provide structure to the project. *”A further purpose of methodologies is to build confidence that the information system will lead to desirable results”* [1]. Therefore an appropriate software development methodology is required to ensure that the project is well structured and to increase the probability of being successful.

There are many different software development methodologies available, each one varying in structure and flexibility, some focusing on technical issues where as others are centred around user involvement. This section evaluates four of the main methodologies used in software development today before deciding upon the most appropriate methodology to be used in this project.

1.3.1 Structured Systems and Design Methodology (SSADM)

SSADM is known as a hard methodology which provides developers with strict guidelines to work with. It is well known for its use in civil service projects and widely used by many organisations. Its compatibility with PRINCE 2, project management methodology makes SSADM preferable over other methodologies.

The five phases of SSADM as outlined in [2] are as follows:

1. Feasibility study
2. Requirements analysis
3. Requirements specification
4. Logical system specification
5. Physical design

Although SSADM is well structured and documented it does not include the later phases of software development, implementation, testing and evaluation.

1.3.2 Systems Development Life Cycle (SDLC)

SDLC is one of the oldest software development methodologies and is also known as ”the waterfall model” comprising of the following phases [3]:

1. Feasibility study
2. System investigation
3. Systems analysis
4. Systems design
5. Implementation
6. Review and maintenance

SDLC is a sequential methodology which does not allow developers to re-do previous phases should they feel that the requirements have changed as it aims to complete the projects on time. SDLC assumes that the development life cycle is short enough to allow another project to start its feasibility study while current project is near finishing phases.

SDLC can be very expensive as it involves repeating of the above phases as the methodology encourages freezing of requirements at the beginning of the life cycle, discouraging change [9].

1.3.3 Rapid Application Development (RAD)

RAD was developed as the need occurred to develop information systems speedily to support continuously changing businesses. It has many similarities with the prototyping approach involving users at early phases in the development process.

As outlined in [2] RAD comprises of the following four phases:

1. Requirements planning
2. User design
3. Construction

4. Cutover

RAD can use third party software and software tools to create working systems in a short period of time. It is important to remember that *”While systems can be built very quickly with RAD tools and methods, system-builders will still need careful requirements planning and modelling to ensure that the systems deliver long-term benefits”* [9]. RAD aims to complete implementation of a system within a 90 day period [2].

1.3.4 Prototyping

Prototyping is a methodology in which the developers create a mock-up of the system very early in the process. It is based upon the belief that *”by interacting with the prototype, users can get a better idea of their requirement”* [9].

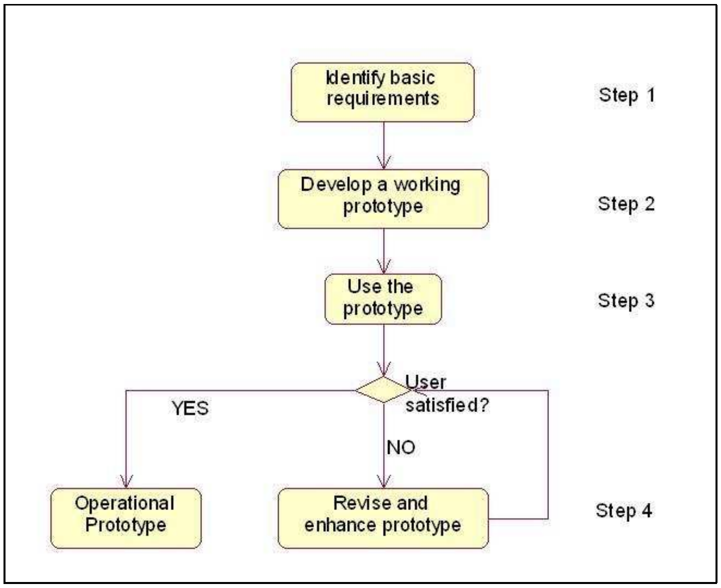

Identify basic requirements
Step 1
Develop a working prototype
Step 2
Use the prototype
Step 3
User satisfied?
YES
NO
Operational Prototype
Revise and enhance prototype
Step 4

Figure 1.1: [9]

Figure 1.1 shows the four phases involved in the Prototyping model. The prototype is modified until the user is completely satisfied. Unless the process is managed, too much time could be spent on prototyping [2]. Prototyping does not include essential phases of the development process making it inappropriate for highly structured projects.

1.3.5 Appropriate methodology

SSADM is not really suitable as its structured nature lacks flexibility which is needed to complete this project in the time available. It also does not include the implementation, testing and evaluation phases of the process which is needed to complete this project. Although SSADM is compatible with PRINCE project management methodology which would make it ideal for large projects it is not necessary for a project of this size.

RAD could be used in the development of this project as it has a high level of user involvement at the early phases of the process. The recommended time period for a project using RAD is much less than time available for this project and as a result the quality of the solution could be affected as not enough time is spent gathering the user requirements. Also RAD does not include all the phases from analysis to evaluation of the development process making it inappropriate for this project.

SDLC seems to be the most appropriate methodology for this project as it includes all the phases from analysis to evaluation of the development process and is a well tested methodology. Although SDLC does not allow the developer to re-do previous phases in the development process should the requirements change, which does not affect this project as the time available will not allow that anyway. As the system is to replace existing non user-friendly solutions, getting the user requirements right for this project is essential. SDLC does not involve the user until the later phases of development which could hinder the success of the system if the requirements are wrong.

Prototyping allows user involvement from early phases but as prototyping does not include the essential phases of the development process, the methodology on its own may not be appropriate as it is equally important that the structure of the system is adequate.

Methodology Engineering (ME) is the process of merging two or more methodologies together to support information systems development [3]. The advantage of ME is that you can *"capture best practice by including techniques in other methodologies, or take into account newly proposed ones"*[2].

Therefore using Methodology Engineering to merge Prototyping with SDLC would seem the most appropriate choice to achieve the best results. Although SDLC methodology will be used for the development process, prototyping will be used for parts of the system that will interact with the user e.g. graphical user interface (GUI).

Based on the phases of the SDLC, the development process will be structured as follows, with appropiate chapter of this report shown in brackets:

1. Feasibility study and System investigation (Introduction)
2. System analysis (Analysis)
3. Systems design including prototyping (Design)
4. Implementation (Implementation)
5. Systems testing (Implementation)
6. Evaluation (Evaluation)

A decision is made to combine the feasibility study and system investigation stages of the SDLC model into one stage as they are very closely linked for the purpose of this project. The last phase of the SDLC model is review and maintenance. Maintenance is required to make sure that any necessary changes are implemented should the requirements have been wrong [3]. Since prototyping is used at the early stages of the process to make sure the user requirements are correct this aspect of the last phase of SDLC will not be required and not possible in the time allocated for this project. As maintenance is not necessary it just leaves review in the last phase of SDLC model, a decision has been made to re-name this evaluation as it is more appropriate for this project. In order to successfully evaluate the solution it is important that the system is tested against the system requirements, systems testing has been added to the SDLC model before the evaluation phase for this purpose.

Prototyping has been added to the systems design phase. After a working prototype has been developed if the user is not completely satisfied the requirements can be changed by going back to the Systems analysis phase.

As well as using SDLC methodology with prototyping, this report will use parts of the Unified Modelling Language (UML). Although UML is known for its use in object orientated system development it can be used with any systems development methodology. ”In UML there are a number of concepts that are used to describe systems and the ways in which they are broken down and modelled” [4]. Using UML diagrams will be particularly useful when carrying out the systems analysis phase of the SDLC methodology to help break down and model the system.

1.4 Main Objectives

The objectives of this project are as follows:

1. Investigate the problem and existing solutions.
2. Gather the user requirements.
3. Select an appropriate methodology and tools to produce a solution.
4. Create prototypes of the Graphical User interfaces (GUI) before producing the solution to ensure it meets the user requirements.
5. Implement a Database and scripts to support the GUI.
6. Test and evaluate the solution.

1.5 Minimum Requirements

The minimum requirements are as follows:

1. A web based system for payroll queries.
2. The system will allow managers to submit and retrieve replies to structured payroll queries.
3. The system will allow payroll personnel to view and reply to payroll queries in a structured format.
4. The system should have a well presented Instruction Manual.

1.6 Additional Requirements

In addition to the minimum requirements, for the solution to address all areas of the problem domain and user requirements it should include some of the additional requirements stated below:

1. The system shall keep a history of payroll personnel interaction with a query which can be viewed by any payroll personnel and the payroll manager.
2. A feature allowing payroll personnel to add comments to a query and view comment associated with a query.
3. Validation ensuring that all the required details needed to resolve the query are present before the manager is allowed to submit a query.
4. Rules implemented to ensure the queries are resolved in order of submission.
5. A Graphical User Interface allowing the payroll manager to count the number of queries matching the criteria stated to compile their report and to view the performance of each payroll personnel.
6. Security measures preventing any user other than payroll personnel and payroll manager to view and reply to queries.

1.7 Deliverables

The following deliverables are derived from both the minimum and additional requirements; some of the deliverables may include more than one of the requirements.

1. Software.
2. Instruction Manual.
3. Final Report.

1.8 Project Schedule

After choosing the appropriate methodology for this project (SDLC with prototyping) it is important that the project is completed on time and that the deliverables are produced.

Accompanied by the Gantt chart in appendix D the table below shows the proposed schedule for completing the phases in the software development methodology used in this project as well as the major deliverables that are not included in the methodology.

The other deliverables form part of the different phases of the software development methodology and are therefore not stated in the project schedule e.g. Prototypes are completed during the design phase and the implemented database and graphical user interface are completed during the implementation phase.

Start Date	Compl Date	Task / * Deliverable
24/10/2004	20/11/2004	Analysis
21/11/2004	19/12/2004	Design
	09/12/2004	* Mid-Project Report
23/01/2005	12/02/2005	Implementation
13/02/2005	26/02/2005	Testing
27/02/2005	19/03/2005	Evaluation
08/03/2005	23/04/2005	Report Writing
	27/04/2005	* Final-Project Report

The project plan is mainly adhered to. The only minor revision is the Analysis phase which is completed a little earlier than scheduled. This allows more time in the design phase to learn a new scripting language.

Chapter 2

Analysis

As described in [3] it is important to establish the requirements of the existing solutions and the problems experienced with them in the Systems investigation stage of the SDLC. *"Identifying what a system should be able to do is one of the first steps in its development"*[4]. This chapter goes through the process of requirements gathering. As SDLC is a sequential methodology which does not allow developers to re-do previous phases should the requirements be wrong, it is important that the user requirements are correct before being frozen at the analysis stage.

This chapter starts by looking at the existing solutions and their problems. It then proceeds to look at requirements gathering techniques and focuses on interviewing. The aim of the interviews was to gather the requirements of the system by looking at the problem domain and discussing the problems of the existing solutions with the three main users and getting them to specify what functionality the new system should have. This chapter concludes by formalising the requirements of the proposed system.

2.1 Existing Solutions

There are three main existing solutions used for processing payroll queries within a large organisation. These are email, telephone and the window approach.

Email

The majority of queries are processed using email. This solution works and to a certain extent meets some of the requirements that need to be addressed in the problem domain. A manager sends an email containing the query to a central address that can be accessed by all payroll personnel. The payroll personnel then check the central folder for emails and reply using their own email account before putting the query in a sub-folder of the central folder known as ’resolved’.

A major problem experienced with this system is that queries are received without containing the minimum required information needed to process it which means that the payroll personnel would then have to reply asking for that information. Also queries are often accidentally misplaced into different folders or deleted both before and after being resolved by payroll personnel. There is no structured way of storing the queries making it difficult to retrieve a particular query without reading each one. A lot of valuable time is wasted by the payroll manager who has to manually count queries for each of their statistics on the reports. Although the system orders the emails by arrival time it does not make sure that this order is taken when resolving them. The email system does not provide a way of tracking history of a query and seeing who is dealing with it.

Telephone

Although the majority of queries are processed using email, there are still some that are dealt with over the telephone. This approach is often taken by the Manager when they feel the query is urgent as it can sometimes take a few hours for the payroll personnel to resolve queries that are received via email. A manager telephones the payroll department and speaks to one of the payroll personnel who would resolve the query instantly and manually record the details of the query on a paper form for statistical purposes.

Although this approach ensures the payroll department has the necessary details to process the query they have to waste valuable time manually recording this information on a paper form. Again this means that the payroll manager would have to manually read the forms to gather the necessary information for their reports. Telephone calls often distract payroll personnel who might have already been trying to resolve queries received via email having an impact on the time taken for queries to be resolved. Telephone queries mean that they have two separate storage solutions for queries making it difficult to find queries that have been resolved in the past.

Window approach

Although the majority of queries are processed either using email or the telephone, there are still some that are dealt with using *"the window approach"*. The window approach is the more traditional one, where the manager would come to the window in the payroll office. This approach is often taken by the Manager when they feel the query is either urgent as it can sometimes take a few hours for the payroll personnel to resolve queries which are received via email or too complex to be dealt with over the telephone or email. A manager would come to the payroll department and speak to one of the payroll personnel face to face, who would resolve the query instantly and manually record the details of the query on a paper form for statistical purposes.

The window approach suffers from the same problems as the telephone. Valuable time is wasted by the payroll personnel manually recording the query information on a paper form and the payroll manager manually reading the forms to gather the necessary information for their reports. Window queries also distract payroll personnel who may be engaged in other activities reducing the efficiency of the department. The Window approach can only be used by Managers who are based at the same site as the centralised payroll department putting managers at other sites at a disadvantage.

A single solution is to be found to combat the problems experienced in the existing solutions and to make sure that there is only one system for processing payroll queries. The solution should improve efficiency within the payroll department and ensure all management have the same opportunities to submit and retrieve replies to queries.

2.2 Requirements Gathering

[3] outlines several requirement gathering techniques which are as follows:

- Questionnaires
- Searching records and documentation
- Sampling
- Observation
- Interviewing

Questionnaires are not used in the requirements gathering as *"Questionnaires are usually used where similar types of information need to be obtained from a large number of respondents or from remote locations"* [3]. Unfortunately due to the nature of this project it will not be possible to use a large number of respondents and as they are not at remote locations, interviewing has been considered more appropriate.

Searching records and documentation is not really appropriate for this project as the problem domain is specified using observation techniques.

Sampling again is not really appropriate for this project as statistical analysis would not really help in specifying the system requirements and would require a large sample of data which is not available in this project.

Observation is used by the author on two occasions during this project. Observation is used to specify the problem domain and the existing solutions. The author's experience of working in the payroll department of a large organisation is extremely useful in providing a good background to the problem. It could be argued that the author may have a biased opinion making it necessary to include the views of others within the requirements gathering process.

Interviewing is used as the main method for requirement gathering as it is seen as one of the best techniques for establishing and verifying the requirements [3] and is most appropriate due to the small number of system users available. Interviewing allows questions to be followed up where as questionnaires would not.

2.2.1 Interviewing

[4] states *"conducting an interview requires good planning, good interpersonal skills and an alert and responsive frame of mind"*, Bearing this in mind the following guidelines described in [4] are adhered to:

Before the interview an appointment is made with the users, at the same time they are told the approximate duration and content of the interview. As only one of each type of user is available it is not necessary to create an interview schedule. Relevant questions are then planned and are shown in appendix E.

At the start of the interview an introduction is made and purpose of interview stated. Permission is granted by the users for notes to be taken during the interview.

During the interview the semi-structured agenda is stuck to and direction controlled. A wide variety of question types are used to gain the information required. The functionality is specified through use case diagrams and points are clarified between the interviewer and interviewee.

After the interview the interviewee is thanked for their time. An additional interview is not necessary as the system requirements are agreed on in the interview.

There are three main users in this project: the manager, the payroll personnel and the payroll manager, through interviewing:

- The manager (Basharat Hussain) is able to specify how he would like to submit and view replies to the queries.
- The payroll personnel (Hilary Rudd) is able specify what such a system needs.
- The payroll manager (Jacqueline Buckley) is able to give me an insight into how she managed the payroll queries and the sort of reports she produces from them.

The interview notes can be found in appendix E.

2.2.2 Use Cases

"Use cases are descriptions of the functionality of the system from the users perspective" [4].

As it is important that the user requirements are correct at the early stages due to the sequential nature of the SDLC methodology, after analysing the interviews with the users a use case diagram is produced and agreed with each user to show what functionality they require from the system (this can be found in appendix F).

After the individual use case diagrams are agreed with the users they are combined to produce the use case diagram shown in figure 2.1 which shows the functionality that the three users require from the system. It also highlights functionality what is required by more than one user e.g. both the payroll personnel and the manager use the system to view replies to the queries.

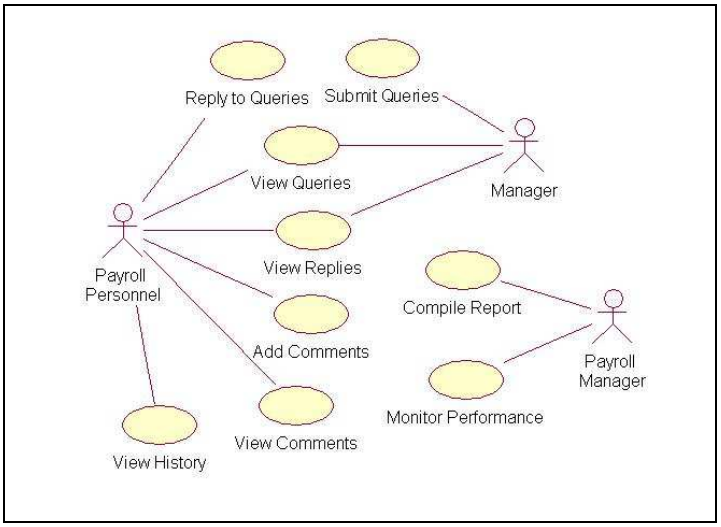


Figure 2.1: Final use case diagram

2.3 System Requirements

The system requirements are divided into two lists, functional requirements and non-functional requirements. Functional requirements are *"what a system does or is expected to do"*[4] whereas non-functional requirements are *"those that describe aspects of the system that are concerned with how well it provides the functional requirements"*[4].

2.3.1 Functional Requirements

1. The system shall allow managers to submit payroll queries.
2. The system shall ensure that all the required details needed to resolve the query are present before submission.
3. The system shall allow managers to view replies to their queries.
4. The system shall allow managers to view their previously resolved queries.
5. The system shall allow payroll personnel to view payroll queries.
6. The system shall store payroll personnel interaction with a query to its history.
7. The system shall allow payroll personnel to add comments to a query.
8. The system shall allow payroll personnel to view the history of a query.
9. The system shall allow payroll personnel to view the comments of a query.
10. The system shall store all payroll queries and replies.
11. The system shall allow payroll personnel to reply to queries.
12. The system shall ensure the queries are resolved in order of submission.
13. The system shall allow the payroll personnel to view payroll queries previously resolved.
14. The system shall count the number of queries matching the criteria stated by the payroll manager to compile her report.
15. The system will allow the payroll manager to view the performance of each payroll personnel.

2.3.2 Non-functional Requirements

1. The system shall allow managers to submit and view replies to queries from any computer with intranet access without having to install additional software.
2. The system shall allow only payroll personnel and payroll manager to reply to queries, view history and comments of a query and view all previously submitted queries.

3. The system shall allow only payroll manager to view the performance of each of the payroll personnel.

4. The system must be easy to use by having a well-designed graphical user interface.

5. The system shall retrieve queries and replies within 30 seconds.

6. The system shall be designed in such a way to allow future development.

7. The system shall support a maximum of 75 simultaneous users (67 managers, 7 payroll personnel and 1 payroll manager)

2.4 Proposed System

The solution to the problem will use a database to store the originally submitted query, any replies, history of the query and data needed for query management. The database will provide central storage for all payroll queries for within the organisation and should reduce time wasted by manually searching for queries using the existing solutions.

The new system should provide the managers with an easy to use graphical user interface (GUI) at which they can submit and view replies to payroll queries for their employees. Existing solutions (email) do not ensure that the manager provides the payroll department with all the necessary information in order to resolve the query, whereas the new system should include rules to prevent this and once this information has been stored it will not need to be duplicated for any other purpose. The managers should be able to receive faster replies to their queries providing the personnel staff use the system effectively as the system will ensure queries are resolved in order of submission.

The new system will provide the payroll personnel with a different GUI that will enable them to view, reply and manage the payroll queries. The GUI interface and database will allow payroll personnel to view the history of any query, add and view comments associated with a query and retrieve previously resolved queries at the touch of a button knowing only one of the attributes of it e.g. the employee's name. The GUI will have additional features allowing the payroll manager to query the database to produce reports. The new system incorporating the above features and the system requirements will save the payroll department a lot of time making it more efficient and reducing costs.

Chapter 3

Design

This chapter starts by exploring a selection of database management systems (DBMS) which are appropriate for use in this project. As efficiency is an important aspect of the proposed system, ER modelling, table design and normalisation is carried out. A number of scripts that run on the web server are needed, these are used by the GUI, one of their purposes is to interact with the DBMS and send data to the GUI. As a result a justification of the choice of scripting language and system architecture is included. The chapter concludes by looking at the HCI issues that need to be considered when implementing the system.

3.1 Database

The solution to the problem will require a Database Management System (DBMS) as a database will be used to store the originally submitted query, any replies, history of the query and data needed for query management.

3.1.1 Database Management System

There are a number of Database Management Systems (DBMS) available; some of the most widely used ones include MySQL, PostgreSQL, MS SQL Server, MS Access, Oracle and DB2. All except

Oracle and DB2 use a Structured Query Language (SQL). *"The primary function of the SQL language is to support the definition, manipulation, and control of data in SQL databases"*[6]. Due to the time available for this project only the DBMS that support SQL are considered. Although MySQL, PostgreSQL, MS SQL Server and MS Access use SQL and can be used with most applications, the DBMS themselves differ making it important that the most appropriate one is used in this project.

3.1.1.1 MySQL

MySQL is open source and as a result has been developed over time making it a very robust and fast DBMS. It is available at no cost which would reduce the cost of the solution if it was used in a commercial environment. Although MySQL is very robust it lacks some advanced features of SQL such as Triggers. *"Because of its limited feature set, MySQL is very fast"*[8].

3.1.1.2 PostgreSQL

PostgreSQL is very similar to MySQL; they are both open source, available freely and are not platform dependent. *"PostgreSQL, on the other hand, provides more features than MySQL"*[8]. The main difference between the two is that PostgreSQL has been designed to handle large Databases and therefore implements some advanced features of SQL such as Triggers. Another advantage of PostgreSQL is that it allows you to implement business logic on the database server, which is useful for developing applications that have highly complex business rules [8]. As a result performance is affected making it slower than MySQL.

3.1.1.3 MS SQL Server

MS SQL Server is designed for very large databases, it can handle a large number of users at any one time and is very easy to maintain through its GUI. Unfortunately it is not freeware and is expensive to purchase. As it is a Microsoft (MS) product it is also platform dependent.

3.1.1.4 MS Access

MS Access has some similarities to MS SQL Server; they are both platform dependent and are easy to maintain through their GUI. MS Access is designed for much smaller databases and desktop applications and as a result do not handle multiple users very well. Another disadvantage of MS Access is that it does not allow remote maintenance.

3.1.1.5 Appropriate DBMS

MS SQL Server could be used as it can handle very large databases and multiple users but due to its expense and platform dependency it is not appropriate for this project as it is important the system could be deployed on any platform. Although MS Access is much cheaper than MS SQL Server as it does not handle multiple users very well it would not be suitable for this project as it is important that a large number of users can use the system at any one time. Both MySQL and PostgreSQL can handle a large number of users making it hard to choose between them. Although MySQL does not implement some advanced features of SQL such as triggers, it is faster than PostgreSQL. As these advanced features are not required for this system and speed is important in this system to improve efficiency of the payroll queries MySQL has been chosen over PostgreSQL.

3.1.2 ER modelling

ER modelling is one of the most common techniques used in database design. The main objective being to *"help in understanding the nature and relationships that exist within the data of the system"*[11]. The ER diagrams can then be used by the designer to derive a logical table schema[11]. The first stage before any ER diagram is drawn is to understand the problem domain[11]. The problem domain for the payroll system is as follows:
Managers submit payroll queries. Payroll personnel view the payroll queries, each viewing being recorded. A query can have comments. Payroll personnel may add comments. Each query will eventually have a reply, a reply is sent by one of the payroll personnel.

Entities are *"any real thing or abstract notion that we wish to recognise as a separate object within the domain"*[11]. Taken from the problem domain the entities for the system are as follows:
Manager, Payroll Personnel, Query, Reply, Comment.

"A weak entity is one which cannot exist without the existence of some other entity"[11]. Taken from the problem domain the weak entities for the system are as follows:
Reply —*as a reply is to a query, a reply can not exist without a query.*
Comment —*as a comment is on a query, a comment can not exist without a query.*

Relationships are *"associations that exist between these entities"*[11]. A relationship has a cardinality that "specifies, for one member of first entity set, the possible number of members it can be related to in the second entity set"[11]. The three possible types of cardinality are listed below along with the relationships that have a cardinality of that type. A relationship also has optionality. A relationship can be; optional, part optional or mandatory. The optionality of a relationship shown below in parenthesis determines whether or not a column attribute can be null affecting the table design.

One to One relationships:

There are no relationships in this problem domain with a one to one cardinality.

One to Many relationships:

- Managers submit payroll queries (part optional).

  One manager may submit many payroll queries (optional).

  One payroll query can only be submitted by one manager (mandatory).

- A query can have comments (part optional).

  One query can have many comments (optional).

  One comment can only belong to one query (mandatory).

- Payroll personnel may add comments (part optional).

  One payroll personnel can add many comments (optional).

  One comment can only be added by one payroll personnel (mandatory).

- A reply is sent by one of payroll personnel (part optional).

  One reply can only be sent by one payroll personnel (mandatory).

  One payroll personnel may send many replies (optional).

- Each query will eventually have a reply (part optional).

  One query can have many replies (optional).

  One reply can only belong to one query (mandatory).

Many to Many relationships:

- Payroll personnel view the payroll queries (part optional).

  One query can be viewed by many payroll personnel (mandatory).

  One payroll personnel may view many queries (optional).

All entities have attributes that describe them. The values of the attributes describe the properties of an instance of the entity[11]. The attributes of the entities in the problem domain are as follows:

Manager —username, password, f_name, l_name.
Payroll Personnel —username, password, f_name, l_name.
Query —ref, f_name, l_name, payroll_no, query, date_time.
Reply —reply, date_time.
Comment —comment, date_time.

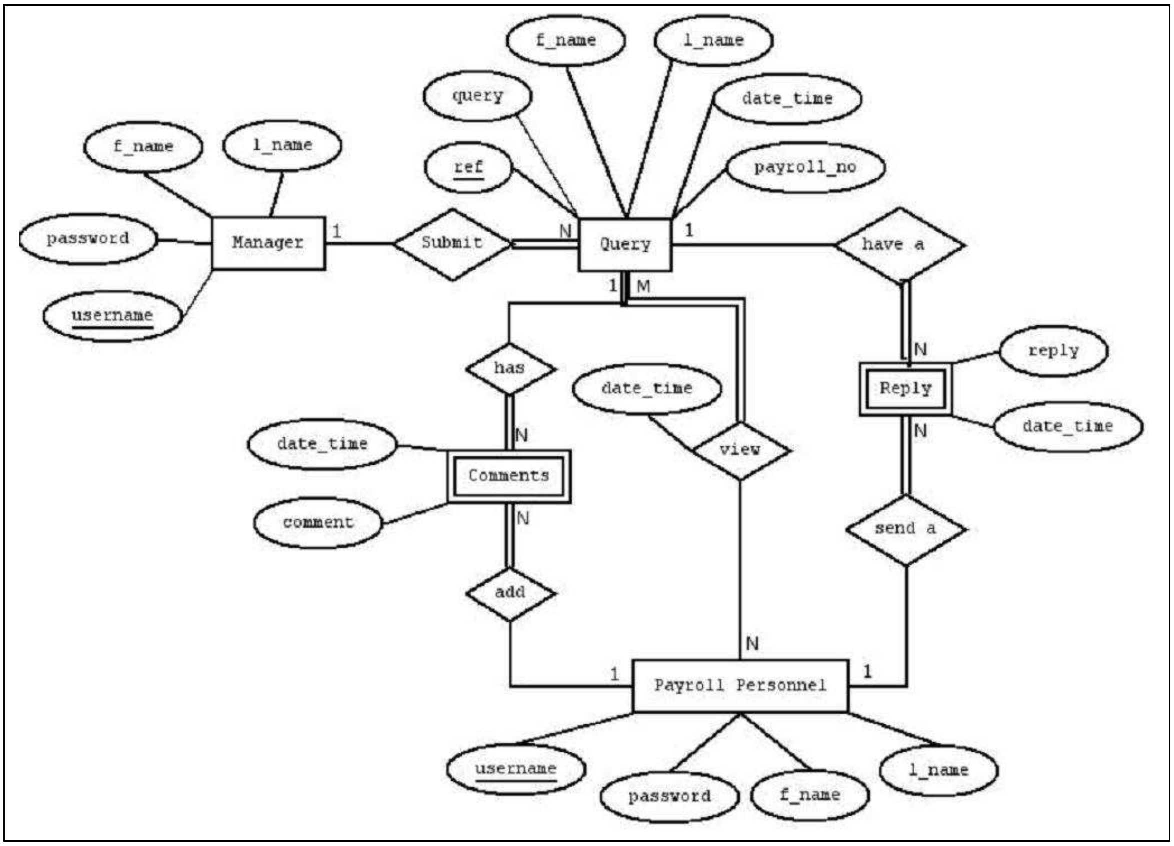


Figure 3.1: Final ER diagram

Figure 3.1 shows the final ER diagram which is derieved from the entities, relationships between the enities, cardinalities of the relationships and the attributes of the entities.

3.1.3 Deriving table design

The following stages described in [11] define the conversion process between the ER diagram and table design:

1. Entity tables —create a table for every entity.
2. Relationships —create new tables or amend those created in step 1.

Entity tables

Attributes of an entity are converted to column attributes of the table. One or more column attributes are chosen as a primary key, the primary key must be unique for each row in the table.

Affected tables with primary keys in bold are shown below:
Manager (**username**, password, f_name, l_name)
Payroll_Personnel (**username**, password, f_name, l_name)
Query (**ref**, f_name, l_name, payroll_no, query, date_time)
Reply (**date_time**, reply)
Comment (**date_time**, comment)

Relationships

One to One relationships
There are no relationships in this problem domain with a one to one cardinality.
One to Many relationships
One to Many relationships is normally represented just using two tables, one table representing each of the entity sets. One of the unique attributes of table one (many end) usually the primary key becomes an additional column attribute in table two (one end) known as a foreign key. The foreign key could form or form part of table two primary key if it helps make a row unique.

Affected tables with primary keys in bold and foreign keys in italics are shown below:
Query (**ref**, f_name, l_name, payroll_no, query, date_time, *Manager.username*)

Reply (**date_time**, ***Query.ref***, ***Payroll_Personnel.username***, reply)
Comment (**date_time**, ***Query.ref***, ***Payroll_Personnel.username***, comment)

Many to Many relationships
Many to Many relationships are represented using three tables, one table representing each of the entity sets and a third table that links rows of the two tables together. The third table primary key is composed of foreign keys which are column attributes of the other two tables usually their primary keys.

Affected tables with primary keys in bold and foreign keys in italics are shown below:
View (**date_time**, ***Query.ref***, ***Payroll_Personnel.username***)

The SQL statements used to create the tables can be found in appendix G.

3.1.4 Normalisation

Normalisation is used when designing relational database tables to ensure that the tables do not contain any anomolies [11]. Following the guidelines described in [11] all tables with the exception of *Query* table are in Boyce-Codd normal form (BCNF). It could be argued that *Query* table would experience the *delete anomaly*, as once a query is deleted information stored about the employee on which the query was based will be lost. In order to make the *Query* table comply with BCNF it would be necessary to create a seperate table *Employee* to store information about the employee. As *payroll_no* field is not mandatory and details regarding employees do not need to be stored for any reason other than the query to which they relate, *Query* table is left in 2NF.

3.2 Scripting Language

There are two types of scripts used in web based systems; a client-side script which executes on the client and is used for setting cookies and validating data before it is sent to the web server and server-side scripts which execute on the web server, interact with the database and *"collect data from the forms and dynamically build Web pages in response"*[13]. Java-script has been used as the client-side scripting language as it is the one most widely used and the one with which the author has the most experience. It is important that a server-side scripting language is found that meets the requirements of the system yet is easy to learn and develop in the time available for this project.

There are a number of server-side scripting languages used in web based systems; the most widely used ones include Hypertext Pre-Processor (PHP), Active Server Pages (ASP), Java Server Pages (JSP) and Practical Extraction and Report Language (PERL). They all differ in structure and ease of development.

3.2.1 PHP

PHP is the most popular server side scripting language because it is easy to develop applications through its large range of libraries and as a result quick to learn. *"PHP usage has exploded as an enterprise application development solution-it's not just for home pages and small projects any more"*[10]. As well as being freeware and platform independent, PHP allows fast integration with MySQL and Apache HTTP Server. PHP creates HTML from the script allowing dynamically created web pages.

3.2.2 ASP and JSP

Working on similar principles to PHP, ASP and JSP are different company's solutions. ASP is a MS product built on Visual Basic (VB) and as a result is platform dependent only working on MS Internet Information Server. Where as JSP has been developed by Sun Microsystems and is built on Java. JSP has the advantage that it can use the large collection of libraries as Java is a more advance language than VB [13]. ASP and JSP due to syntax are a lot harder to learn than PHP.

3.2.3 PERL

PERL used to be the most widely used server-scripting language before being taken over by PHP. It has many libraries for database connectivity but it's not as easy to learn as PHP due to its strong syntax rules. It is often necessary to write complex and tortured code to maintain state in Perl scripts which is not necessary in other scripting languages [13]. Another disadvantage of PERL is that it does not allow HTML to be embedded in the script. It also does not support multi-threading or sophisticated memory data exchange [13].

3.2.4 Appropriate Scripting Language

As apache is the most widely used HTTP (web) server using ASP that is only supported by MS Internet Information Server would be a big mistake as it could restrict the usage of the solution in many organisations. Although PERL has many libraries for database connectivity which is essential for this project, it is harder to learn, may involve writing tortured code and as it does not allow HTML to be embedded in the script it is not the most appropriate tool for this project. ASP and PHP are very similar in terms of features although PHP popularity, ease of learning and fast integration with MySQL makes it a more appropriate tool for the server side scripting in this project.

3.3 System Architecture

The system will comprise of three different layers: presentation layer (GUI), application layer (PHP scripts) and the data layer (MySQL server). These three layers can be spread across several machines described using a Client-Server architecture. The two main types of Client-Server architecture used are as follows:

3.3.1 Two-tiered architecture

The two-tiered architecture consists of two types of machine: clients and servers [14]. The three layers can be divided across the two machines in different ways as shown in figure 3.2.

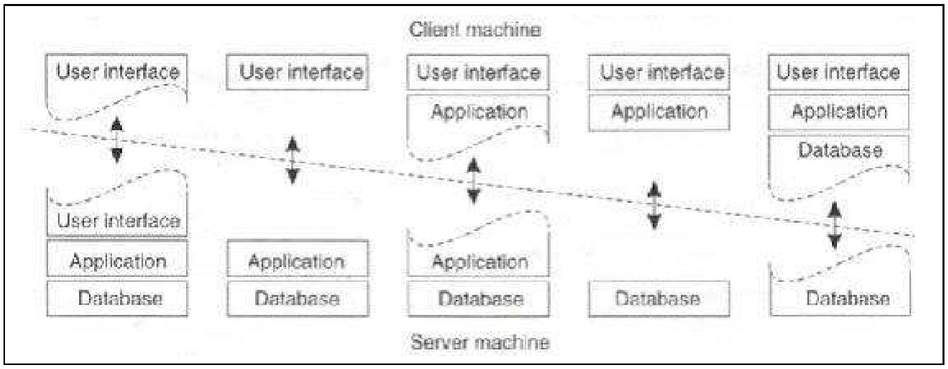


Figure 3.2: [14]

One of the requirements of the system is *"The system shall allow managers to submit and view replies to queries from any computer with internet access without having to install additional software"*. This would mean that system would be limited to using the first two-tiered architecture organisation; only the presentation layer (web browser) on the client, leaving the server with both the application and data layer. As the server would have to do most of the processing a large number of con-current users would affect performance making the two-tiered architecture unsuitable for this project.

3.3.2 Three-tiered architecture

The three-tiered architecture consists of three types of machine organised vertically. It is *”achieved by placing logically different components on different machines”*[14]; client machine (presentation layer - web browser), application server (application layer - PHP scripts) and database server (data layer - MySQL server).

As the processing is divided over two machines; the application server and data server and not just left to one machine as in the two-tiered architecture, the three-tiered architecture is a lot more scalable. The three-tiered architecture can handle more con-current users without affecting performance making it more suitable for this system as there could be up to 200 people using the system at any one time.

3.4 HCI Usability

As the proposed system aims to improve efficiency of payroll queries within large organisations it is important that the system is usable. *”Usability is being able to do the things you want to, not the things you have to!”*[5].

The following characteristics used to measure usability outlined in [5] are considered during the design of the system:

Learnability - ease the system is to learn to use. Although accompanied by a well presented instruction manual, the system should be easy to learn. This could be achieved using a logical navigation structure and page layout.

Efficiency - productiveness of using the system. As well as the three-tiered client-server architecture that should enable the system to handle many concurrent users, the system should have a logically designed

GUI enabling each user to use the system productively.

Memorability - ease to remember how to use the system. Making the system easy to learn will also make the system easy to remember how to use. If the navigation structure and page layout is logical it is a lot easier for the user to picture it in their mind and therefore remember it.

Errors - amount of errors that occur during use. Using java-script (client-side scripting language) the system will validate any data during input reducing the amount of errors that may occur. The use of user friendly error messages will enable users to recover from any error message easily making them feel that they are making positive progress.

Satisfaction - degree to which the users like using the system. Pleasant colours and a logical design, should ensure that the user is satisfied using the system. Having a system that uses dull colours and is hard to navigate is likely to reduce user satisfaction.

Control - amount of control users feel they have over the system. Users like to feel that they are the ones in control. Making sure that the system only does what the user request is essential when designing the GUI.

Skills - degree to which users feel they can utilise their skills when using the system. The system should be professional in the way it interacts with users as it could be used in many large organisations. It should support and enhance the skills of the users and make them feel valued when using it.

Privacy - amount of protection of information belonging to the user or their clients. To enable the system to be efficient allowing payroll personnel to add comments to payroll queries is essential. For the payroll personnel to use this function they have to be confident that managers will not be able to access these comments and that the information is private.

After a prototype was produced the system was rated against each of the above characteristics. The prototype was then improved using the user comments on the characteristics where the prototype scored low.

Chapter 4

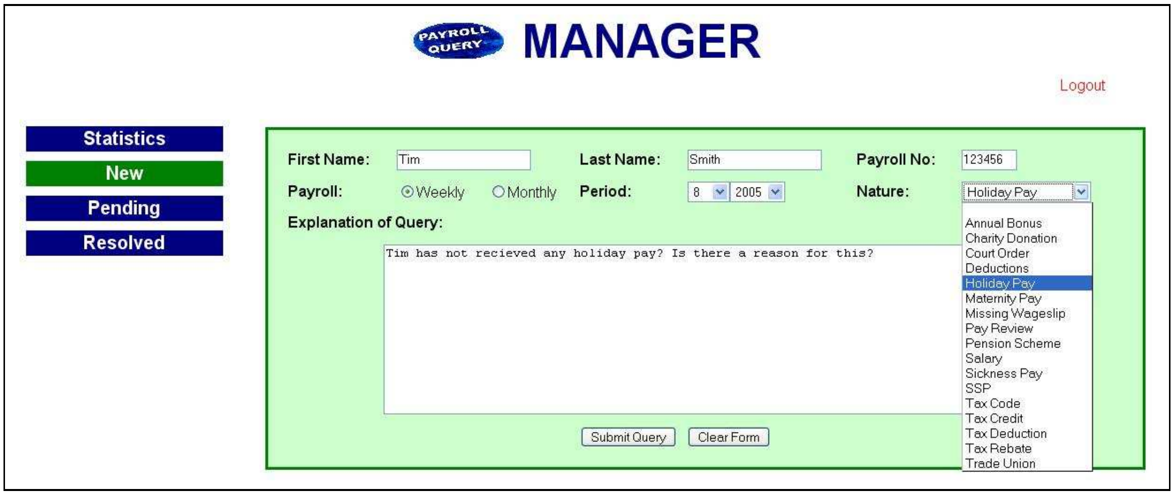
Implementation

This chapter covers the implementation phase of the SDLC methodology chosen in chapter 1. The database design covered in chapter 4 is inplemented and PHP server-side scripts written to create the three tier client-server architecture. No new hardware or software is purchased or installed as the solution is developed on the school's systems. Due to limited space in this report, this chapter takes a brief look at some of the functionality implemented for the three users; manager, payroll personnel and payroll manager. Documentation covering all the functionality implemented in the form of a instruction manual is written as part of this phase and the guidelines followed included in this chapter. The security features implemented and implementation testing are also covered.

4.1 Manager

The Manager interface allows a manager working from any pc with a browser and intranet access to submit, monitor progress and view replies to payroll queries for their employees. The web based solution will enable the large organisation to maintain the system centrally without having to update the software on individual pcs across a number of sites.

The most significant scripts implemented as part of the Manager interface are `manager_new.php`, submit_query.php and submit_query.js. manager_new.php shown in figure 4.1 generates the HTML form used by the manager to enter all the query information. submit_query.js validates the form input ensuring that all information required to resolve the query is present. submit_query.php processes the form input storing the data in the MySQL database.



First Name: Tim Last Name: Smith Payroll No: 123456

Payroll: ● Weekly ○ Monthly Period: 8 2005 Nature: Holiday Pay

Explanation of Query:

Tim has not recieved any holiday pay? Is there a reason for this?

Figure 4.1: manager_new.php

A number of other scipts have been implemented allowing the manager to view statistics regarding the status of their queries and view replies to their submitted queries.

4.2 Payroll Personnel

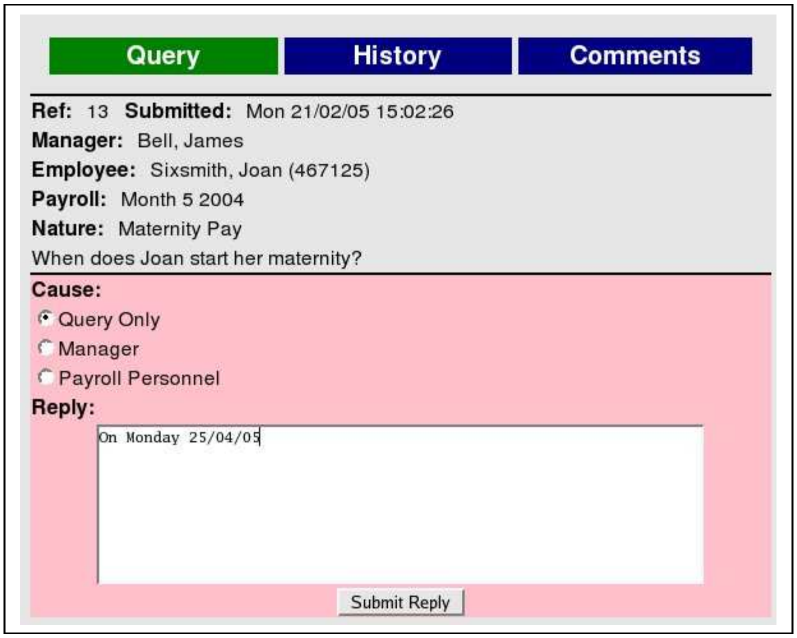
The Payroll Personnel interface is the more complex of the three including the majority of the system functionality. A number of scripts are implemented as part of this interface. Whereas the Manager interface only allows the manager to view their own queries, the Payroll Personnel interface allows payroll personnel to view all queries submitted.

A separate window is opened for each query like that shown in figure 4.2. The window which executes a number of implemented scripts allowing the payroll personnel to:

View and Reply View and reply to the query. When replying the cause must be selected.

View history View history of the query to see which other payroll personnel have viewed the query and added comments.

View and Add Comments View comments added to the query as well as add their own comments to the query.



Ref: 13 **Submitted:** Mon 21/02/05 15:02:26
Manager: Bell, James
Employee: Sixsmith, Joan (467125)
Payroll: Month 5 2004
Nature: Maternity Pay
When does Joan start her maternity?

Cause:
● Query Only
○ Manager
○ Payroll Personnel

Reply:
On Monday 25/04/05

Figure 4.2: `p_p_vqr.php`

4.3 Payroll Manager

The Payroll Manager interface is the less interactive of the three. Composed of a number of scripts using SQL queries to query the MySQL database presenting the results in both tabular and chart form. Defaulting to the current week, the interface allows the payroll manager to change the week on which they require statistics.

The statistics implemented include:

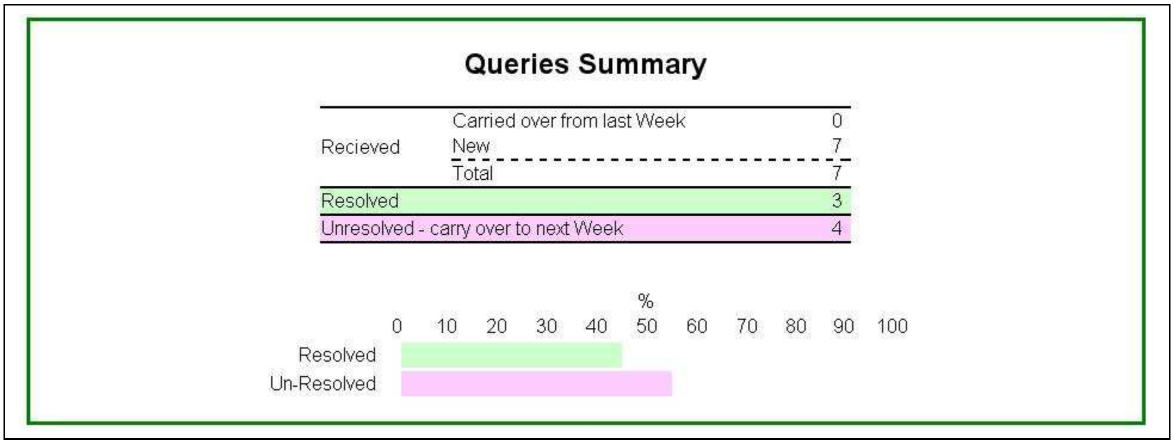
Summary The amount of queries processed by the payroll department (shown in figure 4.3).

Submitted The amount of queries submitted by each manager.

Resolved The amount of queries resolved by each payroll personnel.

Cause Cause of resolved queries, split into three sections; a summary, by manager and by payroll personnel.

Nature Nature of queries submitted.



Queries Summary

Recieved	Carried over from last Week	0
	New	7
	Total	7
Resolved		3
Unresolved - carry over to next Week		4

Figure 4.3: `p_m_main.php`

Additional statistics could easily be added to meet the needs of the organisation in which the system is implemented.

4.4 Security

The new system has a number of security features implemented to ensure that unathorised personnel can not easily view or tamper with what may be confidential data. The new system only allows payroll personnel to reply to queries, view history and comments of a query and view all previously submitted queries. The new system also only allows the payroll manager to view the performance of each of the payroll personnel.

The use of a system login page `welcome.html` which executes the PHP script `authorise_login.php` shown in figure 4.4 enables the system to control access. The authorisation script validates the username and password entered into the system login page establishing the type of user and storing this information in a session variable which is used by the browser until either the user logs out or closes the browser. The session variable is then used by all the other scripts to ensure that the user is authorised to view the information they request.

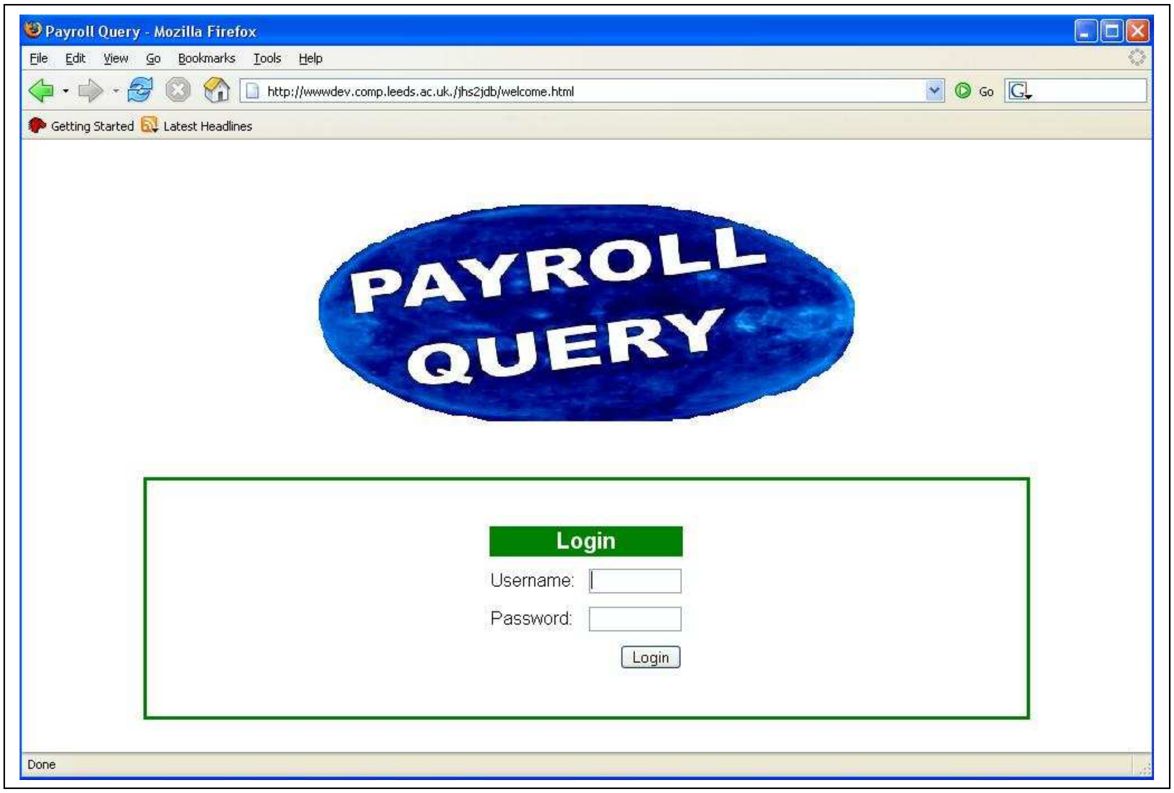
```
<?
  // If username and password have not been provided send back
  if ((!$_POST[’username’]) || (!$_POST[’password’])) {
    header("Location: welcome.html");
    exit;
  }
  // Connect to DB
  include("lib/db_connect.php");
  // Check whether password is valid and a manager
  $sql = "SELECT * FROM manager WHERE username = ’$_POST[username]’ AND password = ’$_POST[password]’";
  $result = mysql_query($sql, $connection) or die(mysql_error());
  $manager = mysql_num_rows($result);
  // Check whether password is valid and a payroll_personnel
  ...
  // Invalid user
  if ($manager == 0 && $p_p == 0){
    header("Location: welcome.html");
    exit;
  }
  else {
    // Start session
    session_start();
    session_register(’username’);
    session_register(’type’);
    // Store username
    $_SESSION[’username’]=$_POST[’username’];
    // Manager
    if ($manager != 0){
      // Store type
      $_SESSION[’type’]="manager";
      // Send header
      header("Location: manager/manager_main.php");
      exit;
    }
    // Payroll_personnel
    ...
    // Payroll_manager
    ...
  }
?>
```

Figure 4.4: `authorise_login.php`

4.5 Testing

The system is developed on *wwwdev* Apache webserver and *csdbdev* MySQL database server. A number of users are set up representing one of the three different types of users for testing. The system functionality is tested throughly after each prototype had been approved by the user. The database currently contains a small set of fictional data that was used during testing and the demonstration.

`welcome.html` shown in figure 4.5 (system login page) can be accessed by entering the following address into the browser `http://wwwdev.comp.leeds.ac.uk/jhs2jdb/` from any computer on the school network.



http://wwwdev.comp.leeds.ac.uk./jhs2jdb/welcome.html

Figure 4.5: `welcome.html`

Fictional users which can be used to test the system are as follows:

User type	Name	username	password
Manager	Adam Brown	browna	1234
Payroll Personnel	Hilary Lucas	lucash	1234
Payroll Manager	John Steele	steelj	1234

Please see the instruction manual (section 4.6) for more information on how to login and use the system.

4.6 Instruction Manual

The instruction manual which can be found in appendix B is created as a step by step guide on how to use the system. The manual is divided into four chapters, the first being an introduction covering how to login and logout of the system. The remaining three chapters cover each of the different types of users. The manual is written following the document quality guidelines described in [12].

Document structure

1. The Manual contains a cover page which identifies the project, the document, the author and the date of production.

2. The Manual is divided into chapters structured into sections and sub-sections using a consistent numbering scheme and page numbers.

3. The Manual contains a contents page which allows information to be found easily.

4. A glossary is not included in this manual as no abbreviations or technical terms have been used. It has been written for the more novice user.

Writing style

1. Active rather than passive tense is used. Eg. ”You will notice that it has been highlighted”.

2. The manual is checked both automatically and by human readers that it contains grammatically correct constructs and correct spelling.

3. Short sentences and numbered lists are used to break down instructions.

4. Paragraphs are kept short to aid memory.

5. All descriptions are short and not verbose.

6. All terms are defined and well explained.

7. No information is referred to by reference number alone, an explanation of what the reference covers is given. Eg. ”see section 3.2.4, this describes how to view the comments added to the query”.

Chapter 5

Evaluation

This chapter covers the final stage of the SDLC methodology, evaluation. Evaluation allows the solution to be assessed. The role of evaluation is to *"test our systems to ensure that they actually behave as we expect and meet the requirements of the user"*[7]. Evaluation is carried out throughout the design phase using prototyping and the implementation phase by testing the functionality of the system. This allowed any problems to be solved quickly before progressing on to the next phase of the methology[7]. The sequential nature of the SDLC methodology and time available for this project made continuous evaluation essential. An heuristic evaluation has been performed assessing system usability accompanied by a statement of satisfaction of functional and non-functional requirements. This chapter concludes by comparing the system with other solutions and looking at future development.

5.1 HCI Usability

HCI principles should be taken into account when designing any system. It is important that the system is usable and that the potential users want to use the system as oppose to having to use it[5].

Heuristic evaluation involves *"experts assessing the design against known usability criteria"* [7]. The usability criteria that the implementation phase of the SDLC methodology should have taken into consideration were defined in the design phase. Heuristic evaluation is used to assess whether the specified usability criteria are successfully applied to the system.

5.1.1 Questionnaire

Questionnaires are used to query the experts and obtain their views on the usability of the system. Although questionnaires are not as flexible as interviews due to their highly structured nature, they take a lot *”less time to administer”*[7]. With time available for this project being scarce, using questionnaires is the more appropriate solution. The experts who completed the questionnaire are from either a computing or payroll background. A balance of both looks at both aspects of usability, a computing expert with their knowledge of other usable systems and a payroll expert based on how usable they find the system.

The heuristics used in the questionnaire are taken directly from the usability criteria specified in chapter four (design phase). The questionnaire aims to assess the eight different areas of usability and asks the expert to answer questions on each area. The areas are as follows: learnability, efficiency, memorability, errors, satisfaction, control, skills and privacy. The questions under the different areas included in the questionnaire are as follows:

Learnability

1. How easy did you find the system was to learn?

2. Did you feel the need to refer to the Instruction Manual? If yes, how often and did it resolve any problems you were experiencing?

3. Did you find the navigational structure and page layout was easy to follow?

Efficiency

4. Did you experience any noticeable delays when using the system?

5. Did you find the page layout supported the order of completing tasks?

Memorability

6. Did you find the system easy to remember how to use?
7. Did you find the navigational structure and page layout supported this?

Errors

8. Did any errors occur whilst using the system? What was the nature of those errors?
9. Were the error messages friendly and did they help you resolve the problem?

Satisfaction

10. Did you enjoy using the system?
11. How did you find the colours and aesthetics of the system?

Control

12. Were there any occasions when you felt you were not in control of the system? If yes, what happened on these occasions?

Skills

13. How professional did you find the system?
14. Did you find it enhanced your skills?
15. Did you feel valued whilst using the system?

Privacy

16. How confident did you feel that the system complied with the data protection legislation practiced in your role?

5.1.2 Conclusions

The answers to the questions posed in the usability evaluation questionnaire can be found in appendix H. Overall the experts found the system to be fairly usable. The conclusions that are derived from the answers under each of the different usability areas are as follows:

Learnability

The experts found the system very easy to learn, the system is uncluttered and that information is easy to find. A minimalist approach makes the system clear and easy to navigate. It seemed that the majority of the experts did not feel the need to refer to the Instruction manual as the system is self explanatory. Those that did, found it useful in explaining more of the payroll query process as oppose to the system itself. The navigation bar on the left made navigation of the system very easy.

Efficiency

It appears that the system is very efficient with only one noted delay from an expert. On this occasion it was the external connection to the schools server that crashed and not the system itself. This should not be a problem when implemented in a large organisation as opposed to externally. The experts found that the page layout and tabulated format supported the completion of tasks.

Memorability

Clarity and a logical structure made the system easy to remember. Functionality consistently located between pages made the system fairly intuitive. The navigational structure and page layout allowed the experts to build a spatial representation of system making it easier to remember. The familiar menu style also helped the experts recall how to use the system.

Errors

Although validation exists on the submit query form to ensure that all details are present before allowing the user to submit the query, there is nothing to ensure that the first and last name fields do not contain numeric data. More validation could be added to the submit query form to ensure data integrity. The experts found that highlighting of incomplete fields is good at alerting the user to missing data, although a more desriptive error message may be useful.

Satisfaction

The majority of the experts enjoyed using the system. The simplicity and ease of use made using the system an enjoyable experience. The colours and aesthetics of the system were easy on the eyes. The colours were found to be modern and clear. A couple of the experts pointed out that the yellow on mouse over for the navigation bar is not the best choice of colour and may not be suitable for the visually impaired. This could be changed with very little work.

Control

The experts generally felt that they were in control of the system at all times. One expert expressed concern regarding the movement of queries from new to pending without asking the user, this is related more to the payroll query process as opposed to the usability of the system. The password system could be improved to provide more feedback as to unathorised access, at present if the username and password are invalid the system denies access without providing any feedback as to why. Again this could be added to the system with a small amount of work.

Skills

The system was found to be professional by the experts, the standardised layout and familiar navigation are the main reasons for this. The payroll experts found that the system either did enhance their skills or would given time. One expert commented that he felt the efficiency of his role would be increased by the system which could impact on his skills. The personalisation features of the system makes the users feel more valued and appreciate the role played within the department.

Privacy

Unawareness of the data protection act (DPA) with some of the experts made feedback in this area less reliable than others. The experts did feel confident that the system complied with the DPA providing the system was deployed within a secure internal network. This would most likely be the case within most large organisations. Extra functionality would need to be added in the long term to delete queries after a certain period to comply fully with the DPA.

5.2 System Requirements

This section evaluates the funtional and non-functional requirements derieved in the analysis phase of the SDLC methodoloy which can be found in chapter 2. After the implementation phase each requirement can be classified as either satisfied, partially-satisfied or un-satisfied. All requirements being satisfied would in the users view be a complete solution to the problem but due to the time available was not possible. This section is broken down into the three different classifications, listing the requirements that fall into each. Satisfied requirements include an explanation of how they are satisified. Partially-satisfied requirements include an explanation detailing the extent to which they are satisfied and what needs to be done to make them completely satisfied. Un-satisfied requirements include an explanation detailing what needs to be done to make them completely satisfied. Partially-satisfied and un-satisfied requirements show areas where the solution could be extended.

5.2.1 Satisfied Requirements

Functional Requirements

1. The system shall allow managers to submit payroll queries.

   This requirement is satisfied using a HTML form (`manager_new.php`) as part of the manager interface which posts data to `submit_query.php`. `submit_query.php` then executes storing the query in the database.

2. The system shall ensure that all the required details needed to resolve the query are present before submission.

   This requirement is satisfied using javascript (`submit_query.js`) to validate the form input data before being posted to `submit_query.php`. The user is informed if data is missing or of incorrect type.

3. The system shall allow managers to view replies to their queries.

   This requirement is satisfied by allowing the user to click on any query listed on the ”resolved” page which opens a new window containing the query and a reply.

4. The system shall allow managers to view their previously resolved queries.

   This requirement is satisfied by the same means as requirement 3. The database stores previously submitted queries indefinatley.

5. The system shall allow payroll personnel to view payroll queries.

   This requirement is satisfied by allowing the user to click on any query listed on the "new", "pending" or "resolved" page which opens a new window containing the query.

6. The system shall store payroll personnel interaction with a query to its history.

   This requirement is satisfied by two PHP scipts. `p_p_vqr.php` *records when the user views a query and* `submit_comment.php` *records comments added to the query.*

7. The system shall allow payroll personnel to add comments to a query.

   This requirement is satisfied by a HTML form (`p_p_add_comment.php`*) within the window for the relevant query which posts data to* `submit_comment.php`*.* `submit_comment.php` *then executes storing the query in the database.*

8. The system shall allow payroll personnel to view the history of a query.

   This requirement is satisfied by `p_p_history.php` *which uses SQL to query both the view and comment tables and generates a HTML page within the query window displaying a list of history of the query.*

9. The system shall allow payroll personnel to view the comments of a query.

   This requirement is satisfied by `p_p_comments.php` *that uses SQL to query the comment table and generates a HTML page within the query window displaying a list of comments added to the query.*

10. The system shall store all payroll queries and replies.

    This requirement is satisfied by the use of a number of HTML forms and PHP scripts using SQL, storing data in a MySQL database.

11. The system shall allow payroll personnel to reply to queries.

    This requirement is satisfied by allowing the user to click on any query listed on the "resolved" page which opens a new window containing the query and a reply.

13. The system shall allow the payroll personnel to view payroll queries previously resolved.

    This requirement is satisfied by the same means as requirement 11. The database stores previously submitted queries indefinatley.

Non-functional Requirements

1. The system shall allow managers to submit and view replies to queries from any computer with intranet access without having to install additional software.

   This requirement is satisfied using a web based solution that requires only a browser to be installed on the client machine. The advantage of using a web based solution is that the system is easier to maintain as updates only need to be made on the server and not on hundreds of client machines across several sites.

3. The system shall allow only payroll manager to view the performance of each of the payroll personnel.

   This requirement is satisfied using a system login page `welcome.html` and session variables. The login page executes `authorise_login.php` which validates the username and password restricting the user to the interface that matches their role, only allowing the payroll manager access to the payroll manager interface.

4. The system must be easy to use by having a well-designed graphical user interface.

   See section 5.1, HCI Usability evaluation.

5. The system shall retrieve queries and replies within 30 seconds.

   This requirement is satisfied, queries and replies can be retrieved straight away. The pages listing the queries update every 10 seconds. This time could of course increase slightly as the number of simultaneous users increases, but should still be lower than 30 seconds.

6. The system shall be designed in such a way to allow future development.

   This requirement is satisfied using well laid out and commented scripts.

7. The system shall support a maximum of 75 simultaneous users (67 managers, 7 payroll personnel and 1 payroll manager).

   This requirement is satisfied, although not explicitly tested. There is no evidence to suggest that Apache webserver and MySQL database server should not be able to handle 75 simultaneous *users. This of course depends on the hardware capacity of the organisation in which the system the system is deployed.*

5.2.2 Partially-Satisfied Requirements

Functional Requirements

14. The system shall count the number of queries matching the criteria stated by the payroll manager to compile her report.
    This requirement is partially-satisfied, a number of statistics pages exist within the payroll manager interface. The statstics required by the payroll manager may differ from one organisation to another and it would be impossible to include all possible statistics in all possible forms in the time availible for this project. The interface is designed in such a way that additional statistics pages could easily be added.

15. The system will allow the payroll manager to view the performance of each payroll personnel.
    This requirement is partially-satisfied, the number of queries resolved by each payroll personnel statistics page exist within the payroll manager interface. Additional statistics could be added showing the performance of each payroll personnel with minimum effort required eg. the average time taken for each payroll personnel to resolve queries.

Non-functional Requirements

2. The system shall allow only payroll personnel and payroll manager to reply to queries, view history and comments of a query and view all previously submitted queries.
    This requirement is partially-satisfied, only payroll personnel currently have access to the payroll personnel interface which contains that functionality. Additional functionality could easily be added to allow the payroll manager to login to the payroll personnel interface as opposed to just the payroll manager interface allowing them to carry out such functionality.

5.2.3 Un-Satisfied Requirements

Functional Requirements

12. The system shall ensure the queries are resolved in order of submission.
    This requirement is not satisfied, no functionality currently exists to implement this requirement.

A number of client-side and server-side scripts would be needed to implement this requirement. Queries would have to be allocated to payroll personnel, allowing others to resolved the next query. The time taken to implement and test this requirement throughly was not availible in this project.

5.3 Achievement of Project Requirements

The minimum requirements of the project are exceeded. A usable web based system for payroll queries has been created, allowing managers to submit and retrieve replies in a structured way. The system not only allows payroll personnel to view and reply to queries in a structured format, extra functionality has been added to allow them to liaise with other payroll personnel to resolve queries more efficiently. A well presented, structured and coherent Instruction Manual can be found in appendix B.

A number of the additional requirements implemented address as much of the problem domain as possible in the time allowed for the project. The system contains functionality that allows payroll personnel to view the history of interaction with a query as well as allowing them to add comments. Validation exists to ensure that all the required details needed to resolve the query are present before the manager is allowed to submit a query. A payroll manager interface allows customisation and compilation of weekly reports accompanied by security measures preventing any user other than payroll personnel and payroll manager to view and reply to queries.

Further information regarding the Implementation of the project requirements can be found in the chapter four (implementation phase) and earlier in this chapter (satisfaction of system requirements).

5.4 Comparison to Other Solutions

The new system has a number of advantages over other solutions. The other main solutions which were discussed in chapter two (analysis phase) are email, telephone and the window approach.

A major problem experienced with email was that queries are received without containing the minimum information needed to process them. The new system avoids this problem by validating the query before submission, ensuring none of the details are left uncompleted. The absence of folders within the new system ensures that queries can not accidentally be misplaced or deleted which was also a concern using email.

A lot of time was wasted by the payroll manager using the other solutions due to the lack of structure and centralisation. The payroll manager would have to manually count all queries matching their criteria wasting valuable time. The new system has a payroll manager interface which contains a number of statistics pages and has been created with the intension of future development and customisation.

Another advantage of the new system is that it allows payroll personnel to manage payroll queries and efficiently liaise with other payroll personnel through history tracking and comment functionality. This functionality should reduce the time taken to resolve queries and is not offered by any of the other solutions.

The new system provides a single, centralised solution, combating the problems experienced in other solutions. If deployed within a large organisation, the system should improve efficiency of the payroll department.

5.5 Future Development

The solution to the problem has a number of areas that could be developed in the future. Development of the system could improve the functionality and customise it to the requirements of the organisation in which it is deployed. Discussed in the previous sections (partially-satisfied and un-satisfied requirements) the main areas for future development are as follows:

- The submission validation could be enhanced to not only prevent the form being submitted without all details being present, but to ensure that all details were of the correct data type. An extension to the existing java script would make this an easy modification.
- Additional statistics pages could be added to the payroll manager interface allowing them to obtain payroll query information that suits their needs. An example which was suggested during the progress meeting would be time taken to resolve queries.
- A more intelligent performance monitoring package could be added to the payroll manager inter-

face. Performance monitoring would allow the payroll manager to take full advantage of information systems when managing payroll personnel, freeing up time for the more skilled tasks.

- Rules could be implemented to ensure the queries are resolved in order of submission. The time available and complexity of this additional requirement meant that it could not be fulfilled in the time available. This would be one of the more challenging functionality to develop in the future.

Bibliography

[1] C Avgerou and T Cornford. *Developing information systems*. MacMillan, Basingstoke, 1993, pp. 166-181.

[2] D Avison and G Fitzgerald. *Information systems development: methodologies, techniques and tools*. McGraw-Hill, London, 2003, pp. 89-94, 379-387.

[3] D E Avison and G Fitzgerald. *Information systems development: methodologies, techniques and tools*. McGraw-Hill, London, 1995, pp. 20-35, 418-424.

[4] S Bennet, S McRobb, and R Farmer. *Object-oriented systems analysis and design using UML*. McGraw-Hill, 2002, pp. 96-105.

[5] J Cato. *User-centered web design*. Addison-Wesley, London, 2001 pp. 1-173.

[6] C J Date and H Darwen. *A guide to the SQL standard*. Addison-Wesley, 1997, pp. 3-26.

[7] A Dix, J Finlay, G Abowd, and R Beale. *Human-Computer Interaction*. Prentice Hall, London, 1993 pp. 363-397.

[8] B Jepson. PostgreSQL vs. MySQL: building better databases. http://www.webtechniques.com/archives/2001/09/jepson/, November 2004.

[9] K C Laudon and J P Laudon. *Management Information systems*. Prentice Hall, New Jersey, 2002, pp. 320-324.

[10] J C Meloni. *PHP - fast and easy web development*. Premier Press, 2002.

[11] C Ritchie. *Relational Database Principles*. YHT Limited, London, 2002 pp. 41-89.

[12] I Sommerville. *Software Engineering*. Addison-Wesley, London, 1992 pp. 571-587.

[13] N Sundarapperuma. Creating dynamic web pages using JSP. http://www.cs.umd.edu/users/walid/fall2003/cmsc424-0101/proj_resources/CreatingJSP.pdf, November 2004.

[14] A S Tanenbaum and M Steen. *Distributed systems: principles and paradigms*. Prentice Hall, New Jersey, 2002 pp. 42-54.

Appendix A

Personal Reflection

One of the first problems I encountered a few weeks into the project was that of ownership. Although the company I work for on a part time basis liked the initial project idea, when we looked into developing the solution on their systems problems started to arise. The security implications and the impact that such a system if implemented incorrectly could have on the business was one problem. The second problem was that all systems development and access to servers was completed at head office which would have meant a great deal of travelling and expense during the implementation phase. This meant that the business would not be able to support the project making the analysis and evaluation phases harder than anticipated. I was very fortunate that my work colleagues were willing to give up some of their own time allowing me to interview them during the analysis phase and complete the usability evaluation questionnaire. I would recommend that students intending to do external projects in the future consider implementation problems that could occur before starting the project.

I am happy with my solution to what I found a genuine problem in the organisation in which I work. Although I was a little disappointed that it was not developed within the organisation, encouragement from my colleges made me feel that it has been a success. Keeping the stakeholders informed as to the progress I was making made them feel valued and wanting to help me during the analysis and evaluation phases of the project. The input from others during these phases was extremely valuable, enabling me to produce a better quality solution and evaluation. Involving others in the project brings new ideas and ensures that more angles to the problem are covered.

When considering which server side scripting language to use during the implementation phase, I was slightly apprehensive about choosing one I had never used before. As I was slightly ahead of schedule, after researching the most appropriate language, I set myself a couple of weeks to learn PHP and built prototypes of the system. I found learning a new language rewarding and feel that my confidence about expanding my computing knowledge in the future has increased.

Following the advice of my supervisor, I started writing up the project report very early which left me plenty of time to concentrate on the later stages without worrying too much. After reading several past projects, I was convinced that Latex was the way forward. Being aware of the learning difficulties associated with it, I decided to give it a go myself. I set myself a day to get to grips with Latex finding Andrew Roberts's tutorial and the newsgroup very helpful. I am pleased with the results and would recommend it to anyone writing such a report.

Setting myself a realistic project schedule at the early stages allowed me to balance the work flow with other modules. I found that I was able to adhere to this schedule, slightly reducing the pressure of final year studies. I would recommend that all future students plan their time realistically as project management is equally important as the solution itself.

I would love to see my solution implemented across large organisations in the future, the analysis and evaluation indicate the advantages that it would bring. The design of the system would allow it to be adapted to suit the needs of an individual business. The project is really only a start showing the impact that Information Technology could have on the payroll department. The system could be further developed to cater for the needs of other or changing processes.

Appendix B

Instruction Manual

A Distributed System for payroll queries within a large organisation

User Manual

John David Bates
2005-02-10

Contents

1 Introduction **1**

1.1 Overview . . . 1

1.2 Logging in . . . 1

1.3 Logging out . . . 2

2 Manager **3**

2.1 Your statistics . . . 3

2.2 Submitting a new query . . . 4

2.3 Your pending queries . . . 6

2.3.1 Viewing . . . 6

2.4 Your resolved queries . . . 8

2.4.1 Viewing . . . 8

3 Payroll Personnel **10**

3.1 Statistics . . . 10

3.2 New queries . . . 11

3.2.1 Viewing . . . 11

3.2.2 Viewing history . . . 12

3.2.3 Viewing comments . . . 13

3.2.4 Adding comments . . . 14

3.2.5 Resolving . . . 17

3.3 Pending queries . . . 18

3.3.1 Viewing . . . 18

3.3.2 Viewing history . . . 18

3.3.3 Viewing comments . . . 19

3.3.4 Adding comments . . . 19

3.3.5 Resolving . . . 19

3.4 Resolved queries . . . 23

3.4.1 Viewing . . . 23

3.4.2 Viewing history . . . 24

3.4.3 Viewing comments . . . 24

3.4.4 Adding comments . . . 24

4 Payroll Manager **25**

4.1 Changing the week . . . 25

4.2 Summary . . . 26

4.3 Queries submitted 28
4.4 Queries resolved 29
4.5 Cause of queries 30
4.6 Nature of queries 33

Chapter 1

Introduction

1.1 Overview

This manual describes how to use all areas of Payroll Query. It is divided into four chapters. The first being this chapter which should be read by all users before proceeding, it covers how to login and logout of Payroll Query. The following three chapters cover each of the three different types of user. Chapter two explains how to use the system as a *Manager*. Chapter three explains how to use the system as a *Payroll Personnel*. Chapter four explains how to use the system as a *Payroll Manager*. You should refer to the chapter that is relevant to your role, once you have logged in to the system you will be directed to the interface that matches your role.

1.2 Logging in

The login window is shown in figure 1.1

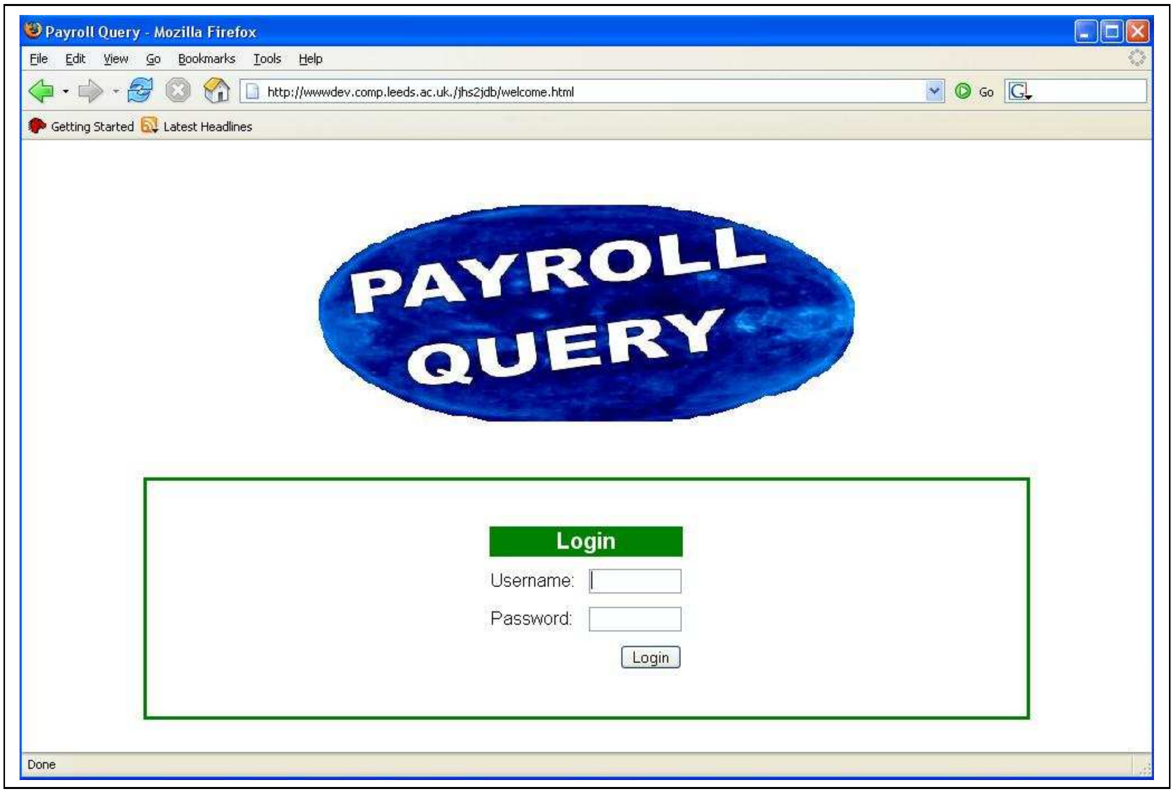

Payroll Query - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://wwwdev.comp.leeds.ac.uk./jhs2jdb/welcome.html
Go
Getting Started
Latest Headlines
PAYROLL QUERY
Login
Username:
Password:
Login
Done

Figure 1.1:

To login.

1. Enter your username into ’Username’ field.
2. Enter your password into ’Password’ field.
3. Left click on ”Login”.

If your password and username do not match those stored in the database or you do not supply a username and password you will not be granted access to the system.

1.3 Logging out

To logout left click on ”Logout” which is located at the top right of the browser similar to that shown in figure 1.2. You will be re-directed back to the login screen where it is safe to shut down the browser if you so wish.

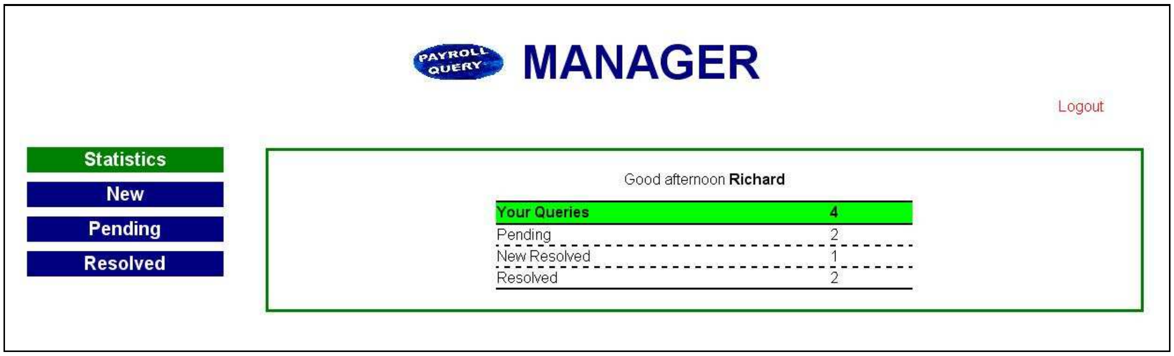
Figure 1.2:

Chapter 2

Manager

The *Manager* interface allows you to view statistics regarding the queries you have submitted, submit new queries, monitor the progress of those queries and view the replies to them.

2.1 Your statistics



Good afternoon **Richard**

Your Queries	4
Pending	2
New Resolved	1
Resolved	2

Figure 2.1:

Statistics shown in figure 2.1 provides information regarding the queries you have submitted.

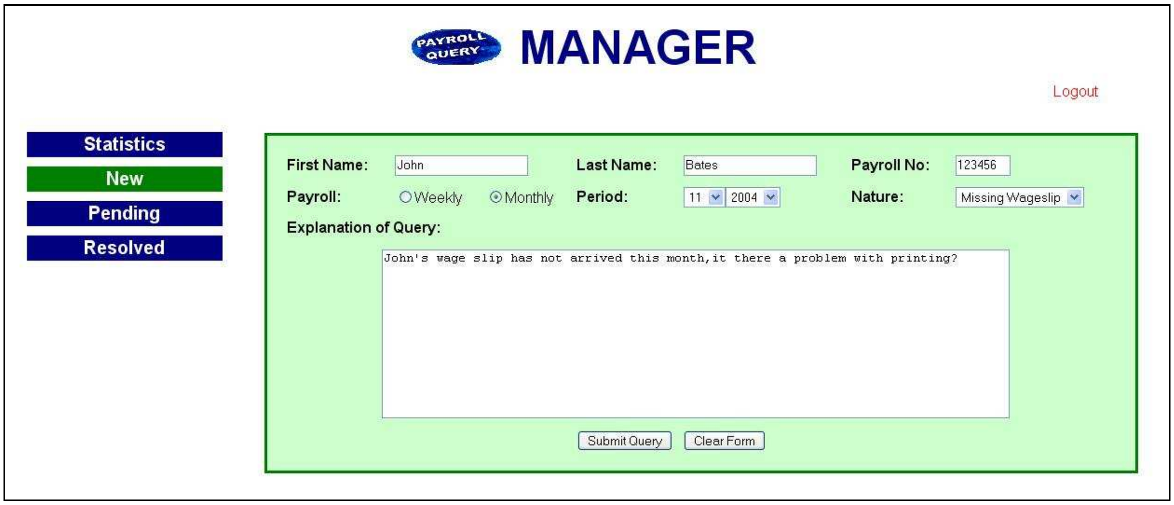
- **Your Queries** is a count of all queries you have submitted.
- **Pending** is a count of all queries you have submitted that haven't yet been resolved.
- **New Resolved** is a count of all queries you have submitted that have been resolved but not yet read by yourself.
- **Resolved** is a count of all queries you have submitted that have been resolved (including those not yet read by yourself).

Statistics can be viewed by left clicking on ”Statistics” located towards the left side of the browser.

2.2 Submitting a new query

To submit a new query.

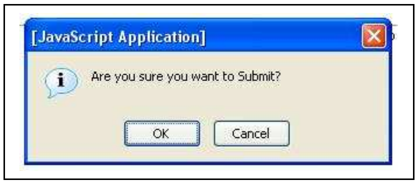
1. Left click on ”New” located towards the left side of the browser.
2. You should see the form shown in figure 2.2.



First Name: John
Last Name: Bates
Payroll No: 123456
Payroll: ○ Weekly ● Monthly
Period: 11 2004
Nature: Missing Wageslip
Explanation of Query: John's wage slip has not arrived this month,it there a problem with printing?

Figure 2.2:

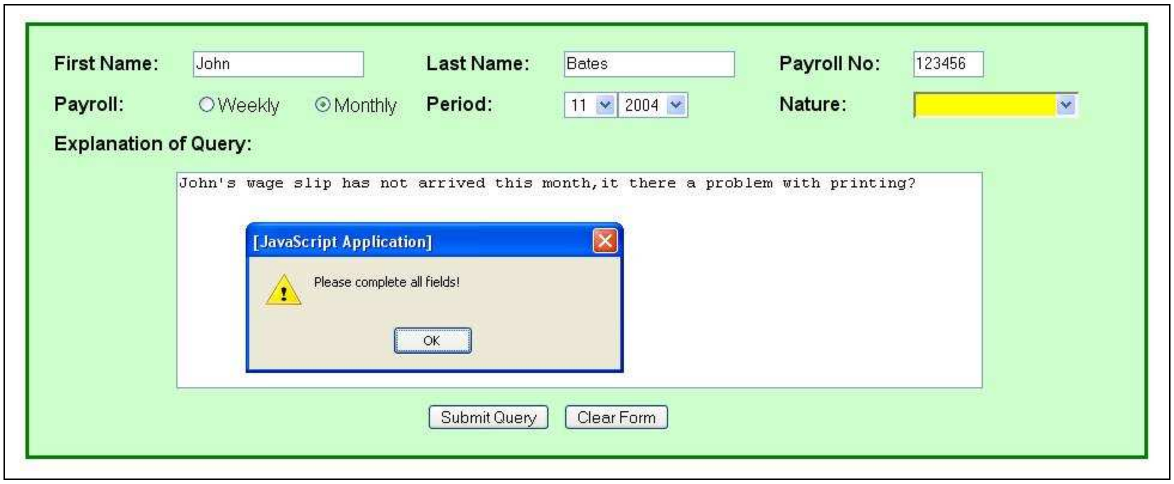
3. Complete the employee’s ’First Name’, ’Last Name’ and ’Payroll No’ fields.
4. Select the payroll which is applicable to the employee (e.g. Weekly or Monthly).
5. Select the period which the query relates to from the left drop down box in the ’Period’ field.
6. Select the year which the query relates to from the right drop down box in the ’Period’ field.
7. Select the nature of the query from the drop down box in the ’Nature’ field.
8. Type the query into ’Explanation of Query’ box.
9. Left click on ”Submit Query”.
10. You should see the box in figure 2.3.
11. Left click on ”OK” if you are sure you want to submit or ”Cancel” if you don’t.



Are you sure you want to Submit?

Figure 2.3:

If you have not completed all the fields or the 'Payroll No' field contains anything other than numbers those fields will be highlighted and the error message shown in figure 2.4 will be displayed.



First Name: John

Last Name: Bates

Payroll No: 123456

Payroll: ○ Weekly ● Monthly

Period: 11 2004

Nature:

Explanation of Query:

John's wage slip has not arrived this month,it there a problem with printing?

[JavaScript Application]

Please complete all fields!

Figure 2.4:

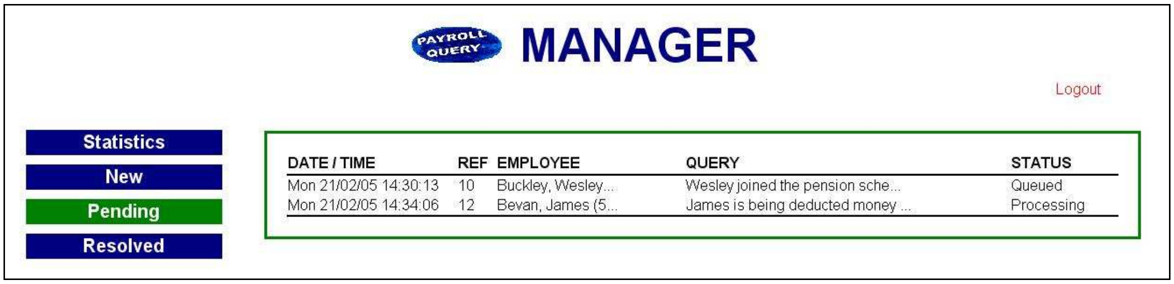
2.3 Your pending queries

Pending queries are those that you have submitted but have not yet been resolved by the Payroll Personnel.

Left click on ”Pending” located towards the left side of the browser. The date/time of submission, the reference number assigned, employees details, the status of query and part of the query are displayed for your queries that are pending.

There are two possible statuses for pending queries (shown in figure 2.5):

- **Queued** indicates the query is waiting in a queue and not yet been looked at by the Payroll Personnel.
- **Processing** indicates the query is being looked at and processed by the Payroll Personnel.



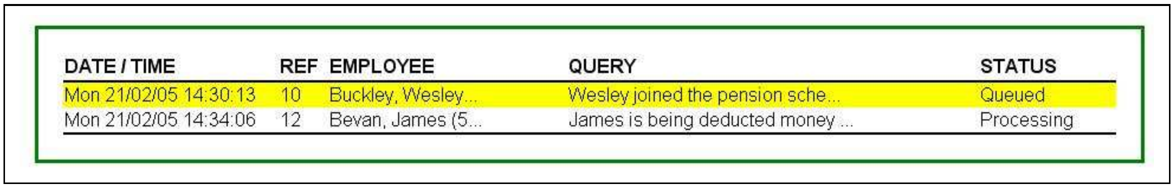
DATE / TIME	REF	EMPLOYEE	QUERY	STATUS
Mon 21/02/05 14:30:13	10	Buckley, Wesley...	Wesley joined the pension sche...	Queued
Mon 21/02/05 14:34:06	12	Bevan, James (5...	James is being deducted money ...	Processing

Figure 2.5:

2.3.1 Viewing

To view a pending query.

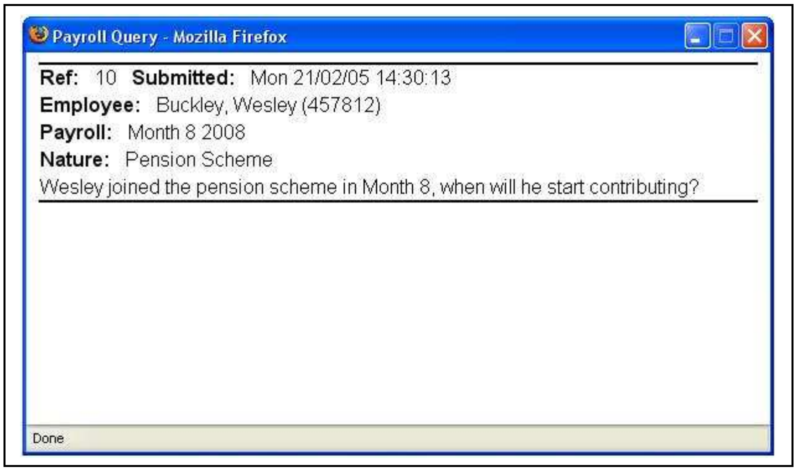
1. Move your mouse pointer over the query you wish to view.
2. You will notice that it is highlighted (shown in figure 2.6).
3. Left click on the highlighted query.



DATE / TIME	REF	EMPLOYEE	QUERY	STATUS
Mon 21/02/05 14:30:13	10	Buckley, Wesley...	Wesley joined the pension sche...	Queued
Mon 21/02/05 14:34:06	12	Bevan, James (5...	James is being deducted money ...	Processing

Figure 2.6:

4. A new window should open like that in figure 2.7.
5. The window will contain the information you submitted in the query.



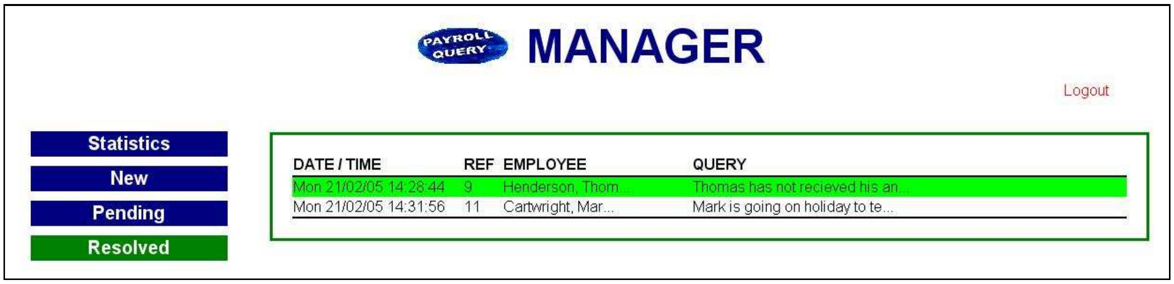
Ref: 10 **Submitted:** Mon 21/02/05 14:30:13
Employee: Buckley, Wesley (457812)
Payroll: Month 8 2008
Nature: Pension Scheme
Wesley joined the pension scheme in Month 8, when will he start contributing?

Figure 2.7:

2.4 Your resolved queries

Resolved queries are those that you have submitted and have been resolved by the payroll personnel.

Left click on ”Resolved” located towards the left side of the browser. The date/time of submission, the reference number assigned, employee’s details and the part of the query are displayed for your queries that are resolved. The resolved queries which you have not read are highlighted in green as shown in figure 2.8.



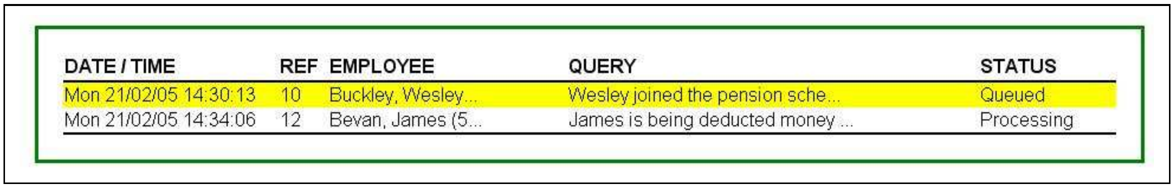
DATE / TIME	REF	EMPLOYEE	QUERY
Mon 21/02/05 14:28:44	9	Henderson, Thom...	Thomas has not recieved his an...
Mon 21/02/05 14:31:56	11	Cartwright, Mar...	Mark is going on holiday to te...

Figure 2.8:

2.4.1 Viewing

To view a resolved query.

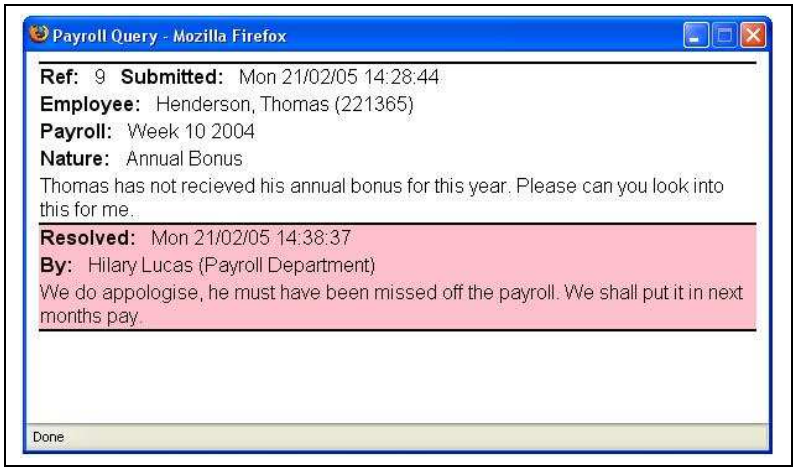
1. Move your mouse pointer over the query you wish to view.
2. You will notice that it is highlighted (shown in figure 2.9).
3. Left click on the highlighted query.



DATE / TIME	REF	EMPLOYEE	QUERY	STATUS
Mon 21/02/05 14:30:13	10	Buckley, Wesley...	Wesley joined the pension sche...	Queued
Mon 21/02/05 14:34:06	12	Bevan, James (5...	James is being deducted money ...	Processing

Figure 2.9:

4. A new window should open like that in figure 2.10.
5. The window will contain the information you submitted in the query along with the reply from the Payroll Personnel highlighted in pink.



Ref: 9 **Submitted:** Mon 21/02/05 14:28:44
Employee: Henderson, Thomas (221365)
Payroll: Week 10 2004
Nature: Annual Bonus

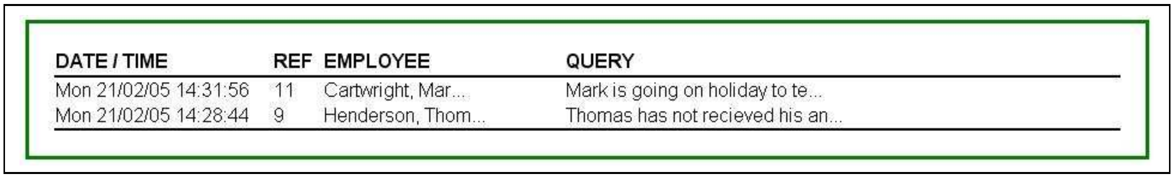
Thomas has not recieved his annual bonus for this year. Please can you look into this for me.

Resolved: Mon 21/02/05 14:38:37
By: Hilary Lucas (Payroll Department)

We do appologise, he must have been missed off the payroll. We shall put it in next months pay.

Figure 2.10:

Once you have opened the resolved query it will no longer be highlighted in green as shown in figure 2.11.



DATE / TIME	REF	EMPLOYEE	QUERY
Mon 21/02/05 14:31:56	11	Cartwright, Mar...	Mark is going on holiday to te...
Mon 21/02/05 14:28:44	9	Henderson, Thom...	Thomas has not recieved his an...

Figure 2.11:

Chapter 3

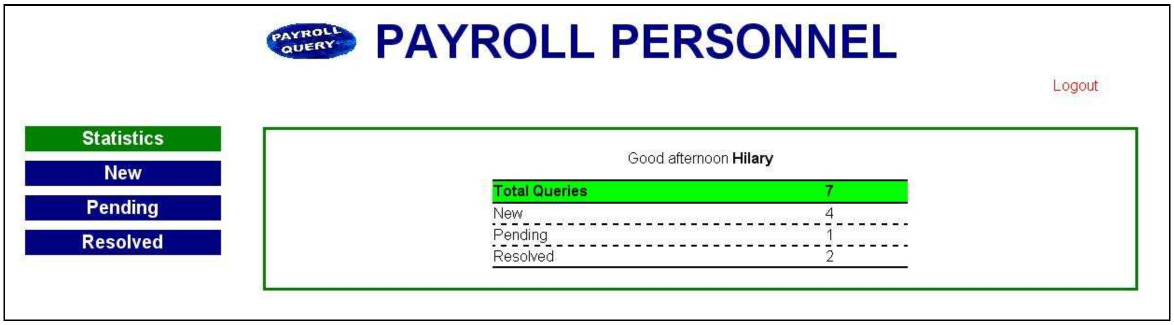
Payroll Personnel

3.1 Statistics

Statistics provides information regarding the queries the Payroll department have received.

- **Total Queries** is a count of all queries received.
- **New** is a count of all queries received that have not been opened by one of the Payroll Personnel.
- **Pending** is a count of all queries received that have been opened by one of the Payroll Personnel.
- **Resolved** is a count of all queries that have been resolved by one of the Payroll Personnel.

Statistics can be viewed by left clicking on ”Statistics” located towards the left side of the browser.



Good afternoon **Hilary**

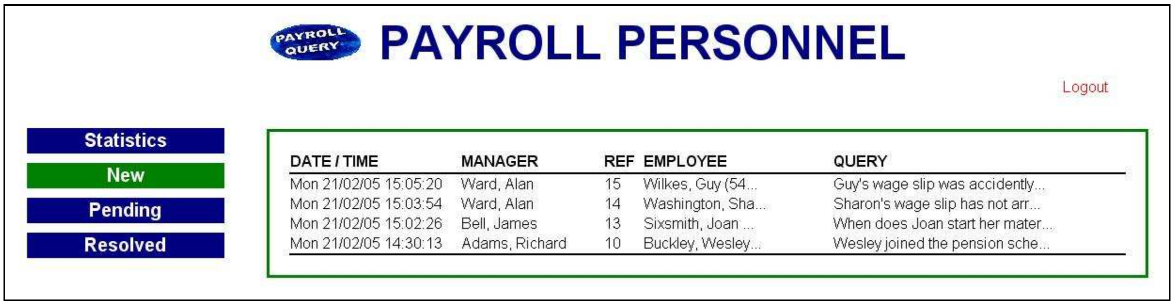
Total Queries	7
New	4
Pending	1
Resolved	2

Figure 3.1:

3.2 New queries

New queries are those that have not been opened by one of the Payroll Personnel.

Left click on ”New” located towards the left side of the browser. The date/time of submission, the manager who submitted the query, the reference number assigned, employee’s details and the part of the query are displayed for the new queries shown in figure 3.2.



DATE / TIME	MANAGER	REF	EMPLOYEE	QUERY
Mon 21/02/05 15:05:20	Ward, Alan	15	Wilkes, Guy (54...	Guy's wage slip was accidently...
Mon 21/02/05 15:03:54	Ward, Alan	14	Washington, Sha...	Sharon's wage slip has not arr...
Mon 21/02/05 15:02:26	Bell, James	13	Sixsmith, Joan ...	When does Joan start her mater...
Mon 21/02/05 14:30:13	Adams, Richard	10	Buckley, Wesley...	Wesley joined the pension sche...

Figure 3.2:

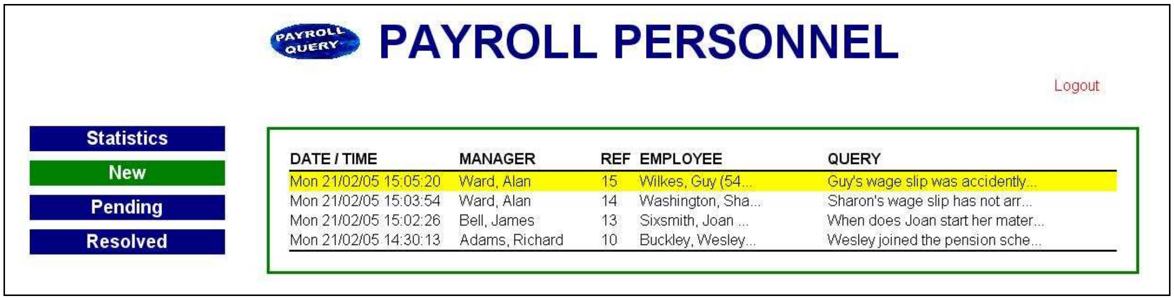
3.2.1 Viewing

To view a new query.

1. Move your mouse pointer over the query you wish to view.

2. You will notice that it is highlighted (shown in figure 3.3).

3. Left click on the highlighted query.

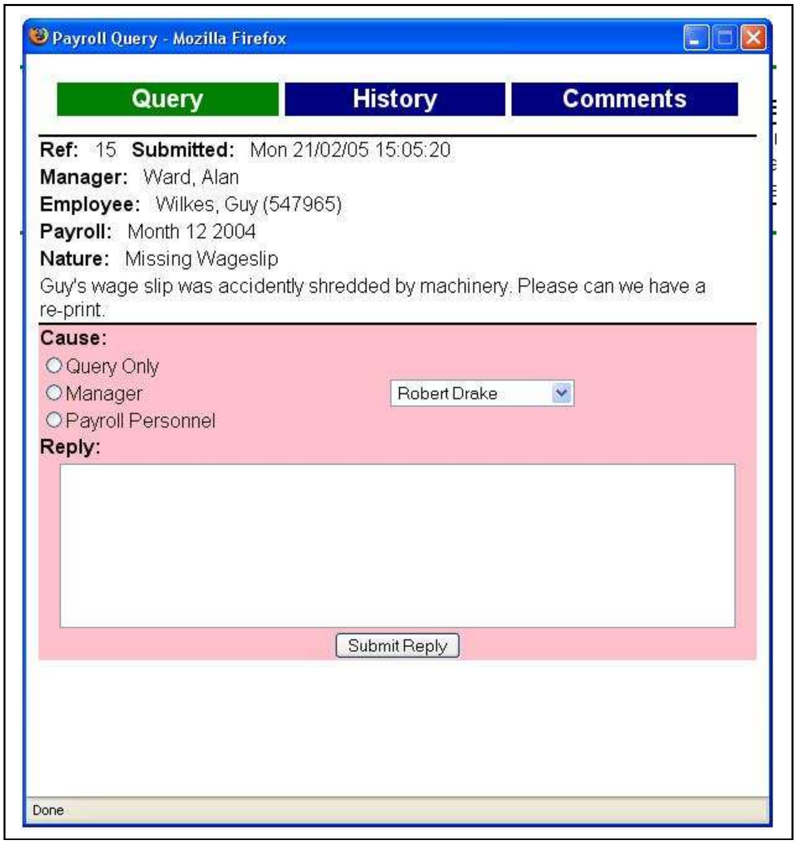


DATE / TIME	MANAGER	REF	EMPLOYEE	QUERY
Mon 21/02/05 15:05:20	Ward, Alan	15	Wilkes, Guy (54...	Guy's wage slip was accidently...
Mon 21/02/05 15:03:54	Ward, Alan	14	Washington, Sha...	Sharon's wage slip has not arr...
Mon 21/02/05 15:02:26	Bell, James	13	Sixsmith, Joan ...	When does Joan start her mater...
Mon 21/02/05 14:30:13	Adams, Richard	10	Buckley, Wesley...	Wesley joined the pension sche...

Figure 3.3:

4. A new window should open like that in figure 3.4.

5. The window will contain the information submitted by the manager in the query along with a reply box.



Ref: 15 **Submitted:** Mon 21/02/05 15:05:20
Manager: Ward, Alan
Employee: Wilkes, Guy (547965)
Payroll: Month 12 2004
Nature: Missing Wageslip
Guy's wage slip was accidently shredded by machinery. Please can we have a re-print.

Cause:
○ Query Only
○ Manager Robert Drake
○ Payroll Personnel
Reply:

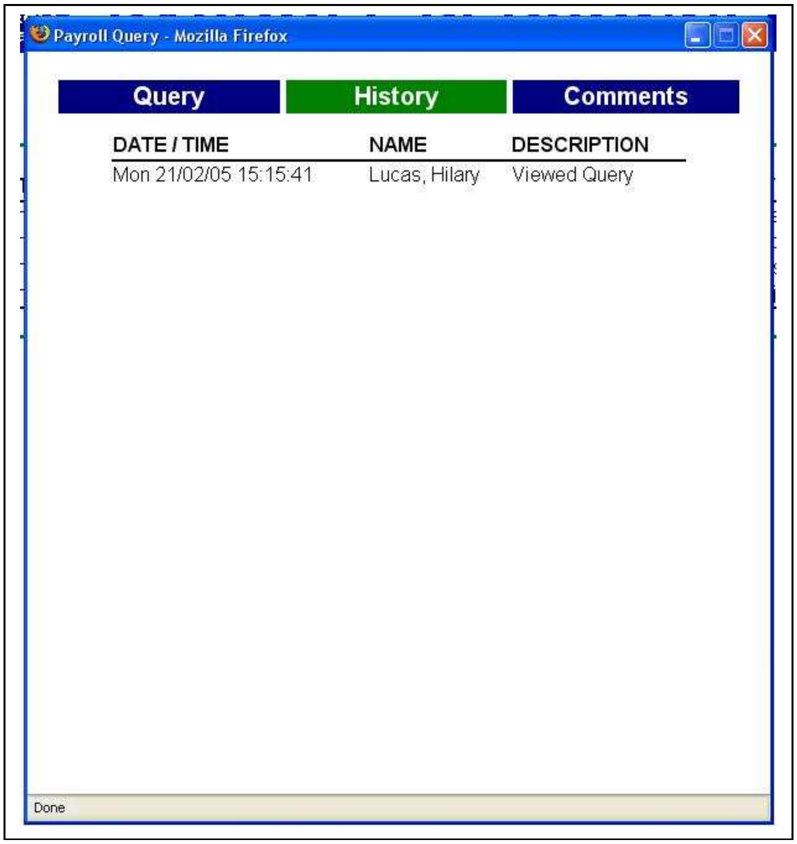
Figure 3.4:

3.2.2 Viewing history

New queries will only show your name in the history as they would be *pending* if someone else had viewed the query previously. See section 3.3 for further information on *pending queries*.

To view the history of the query.

1. Left click on ”History” at the top of the new window shown in figure 3.4.
2. You will see a list showing date/time, name of the Payroll Personnel and a description of what they did with the query as shown in figure 3.5.



DATE / TIME	NAME	DESCRIPTION
Mon 21/02/05 15:15:41	Lucas, Hilary	Viewed Query

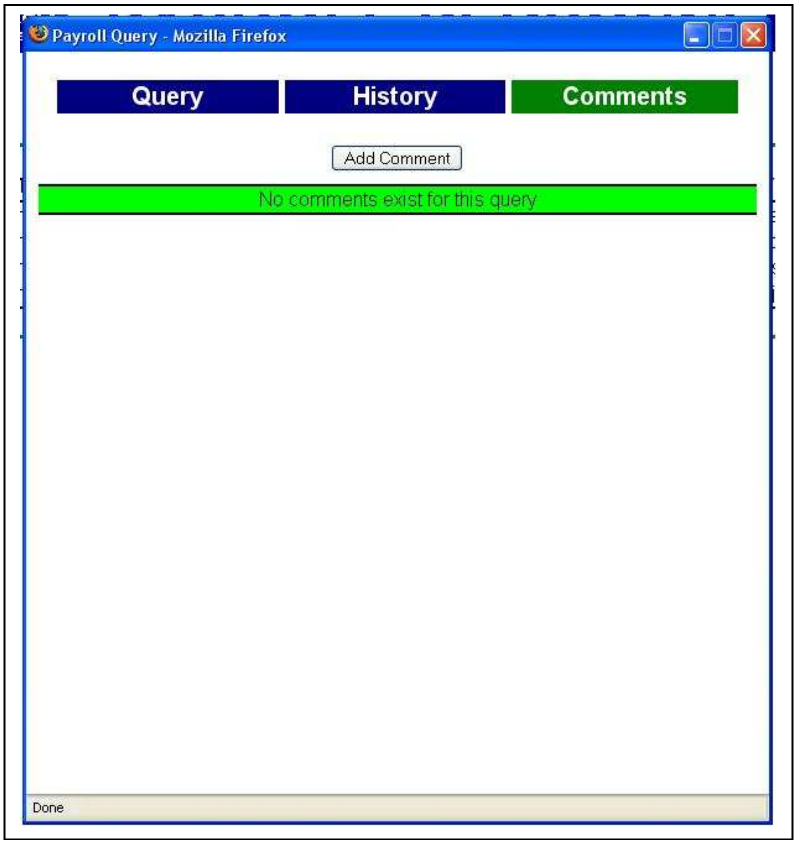
Figure 3.5:

3.2.3 Viewing comments

New queries will not have any existing comments as they would be pending if someone else had viewed and added comments to the query.See section 3.3 for further information on *pending* queries.

To view the comments added to the query.

1. Left click on ”Comments” at the top of the new window.
2. You will see a list showing date/time, name of the Payroll Personnel and a commented they added to the query as shown in figure 3.6.



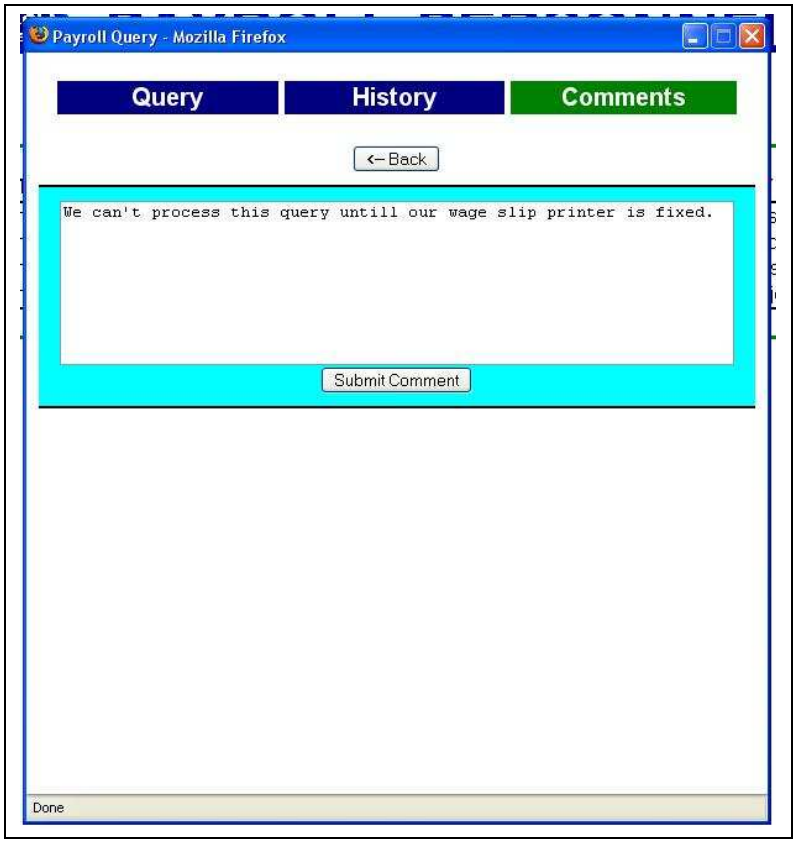
No comments exist for this query

Figure 3.6:

3.2.4 Adding comments

To add a comment to the query.

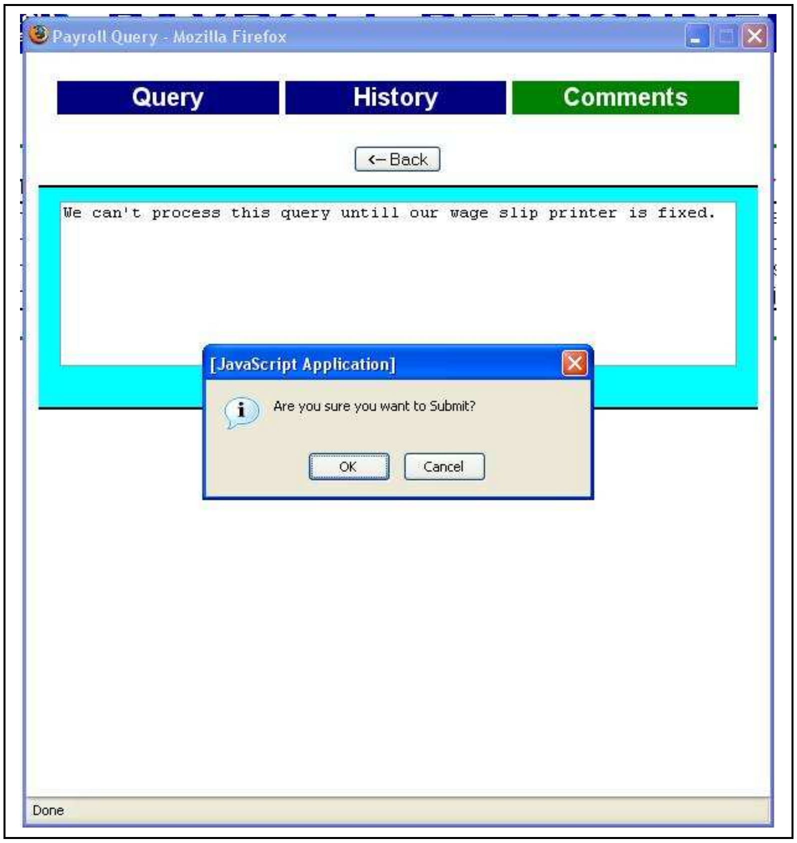
1. Left click on ”Add Comment” shown in figure 3.6.
2. A text box should appear like that in figure 3.7.
3. Type your comment into the text box.
4. Left click on ”Submit Comment”.



We can't process this query untill our wage slip printer is fixed.

Figure 3.7:

5. You should see the box shown in figure 3.8.
6. Left click on ”OK” if you are sure you want to submit or ”Cancel” if you don’t.

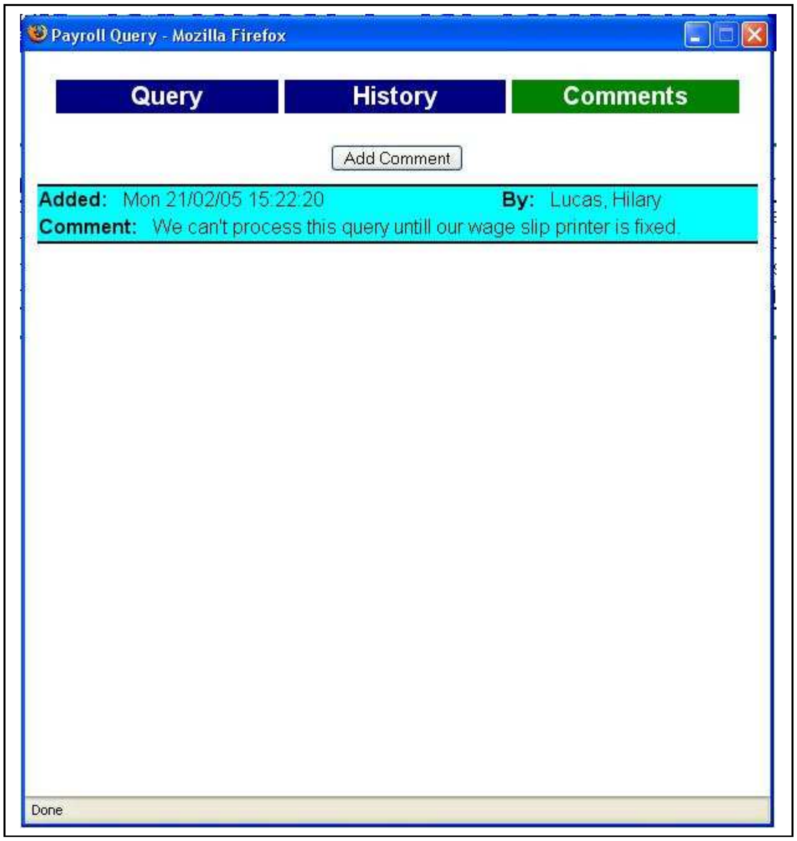


```
We can't process this query untill our wage slip printer is fixed.
```

Are you sure you want to Submit?

Figure 3.8:

7. You should see your comment along with any previously added comments as shown in figure 3.9.



Added: Mon 21/02/05 15:22:20 By: Lucas, Hilary
Comment: We can't process this query untill our wage slip printer is fixed.

Figure 3.9:

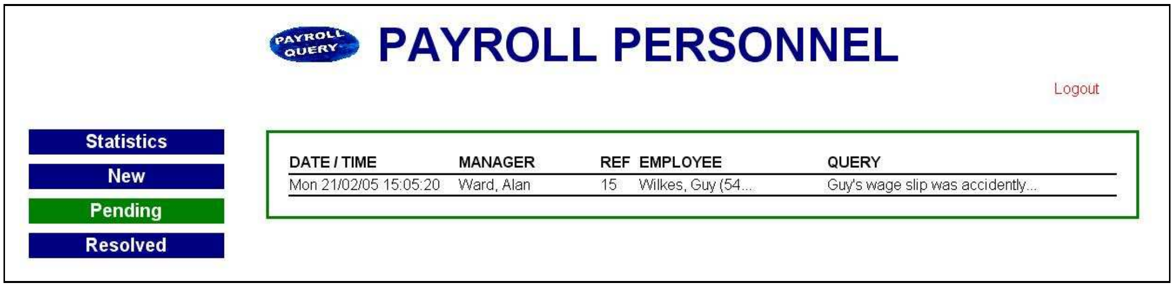
3.2.5 Resolving

To resolve a query see section 3.3.5, this describes how to resolve a *pending* query.

3.3 Pending queries

Pending queries are those that have been viewed by one or more of the Payroll Personnel and may have several comments added.

Left click on ”Pending” located towards the left side of the browser. The date/time of submission, the manager who submitted the query, the reference number assigned, employee’s details and the part of the query are displayed for the *pending* queries shown in figure 3.10.



DATE / TIME	MANAGER	REF	EMPLOYEE	QUERY
Mon 21/02/05 15:05:20	Ward, Alan	15	Wilkes, Guy (54...	Guy's wage slip was accidently...

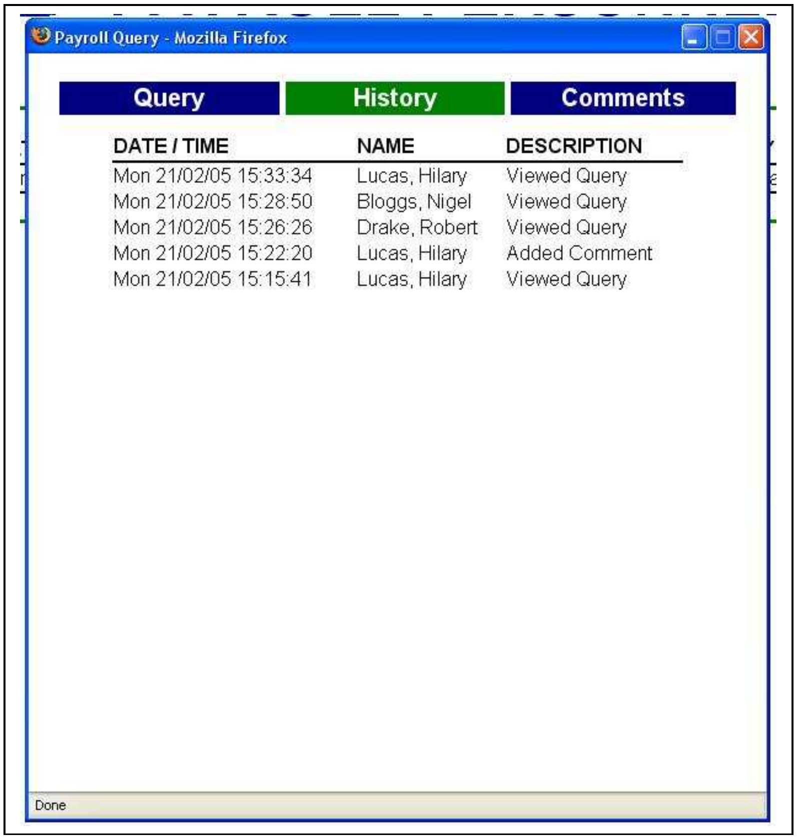
Figure 3.10:

3.3.1 Viewing

To view a *pending* query see section 3.2.1, this describes how to view a *”new”* query.

3.3.2 Viewing history

To view the history of the query see section 3.2.2, this describes how to view the history of the query. The history of a *pending* query may look more like the one shown in figure 3.11.



DATE / TIME	NAME	DESCRIPTION
Mon 21/02/05 15:33:34	Lucas, Hilary	Viewed Query
Mon 21/02/05 15:28:50	Bloggs, Nigel	Viewed Query
Mon 21/02/05 15:26:26	Drake, Robert	Viewed Query
Mon 21/02/05 15:22:20	Lucas, Hilary	Added Comment
Mon 21/02/05 15:15:41	Lucas, Hilary	Viewed Query

Figure 3.11:

3.3.3 Viewing comments

To view the comments added to the query see section 3.2.3, this describes how to view the comments added to the query.

3.3.4 Adding comments

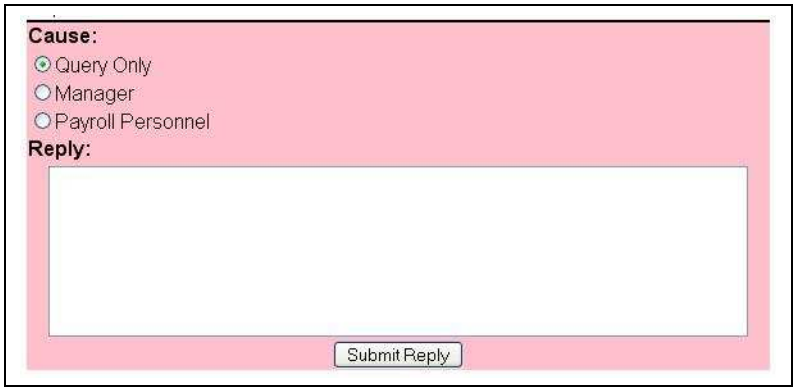
To add comments to the query see section 3.2.4, this describes how to add comments to the query.

3.3.5 Resolving

To resolve a query.

1. Left click on ”Query” at the top of the new window.

2. Select the cause of the query. If no one is to blame for the query select ”Query Only” as shown in figure 3.12. If the query was caused by the manager select ”Manager” as shown in figure 3.13. If the query was caused by one of the Payroll Personnel select ”Payroll Personnel”, a drop down box will appear as shown in figure 3.14, Select the name of the Payroll Personnel from the drop down box.

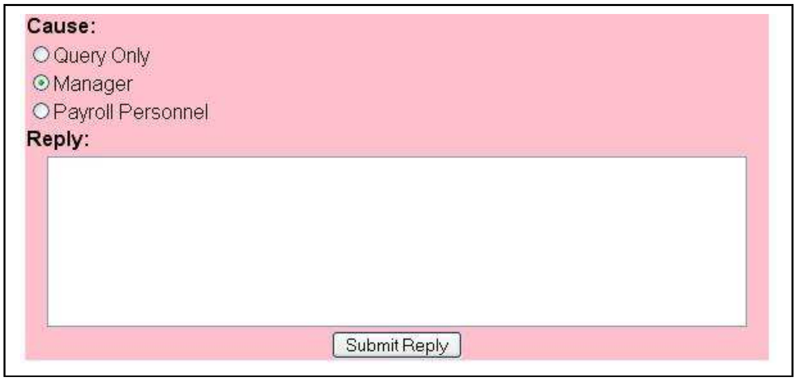


Cause:

● Query Only
○ Manager
○ Payroll Personnel

Reply:

Figure 3.12:

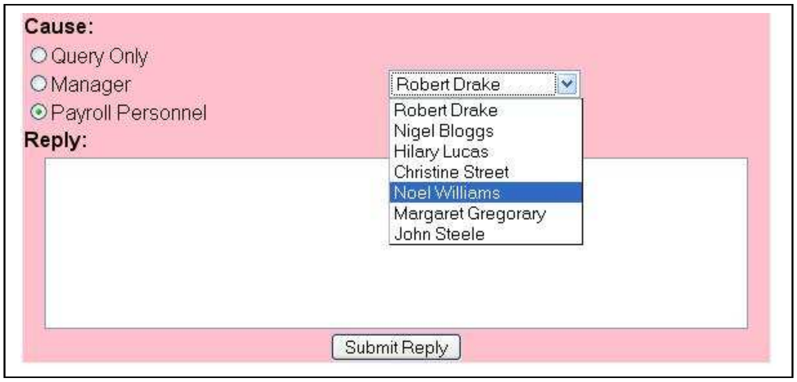


Cause:

○ Query Only
● Manager
○ Payroll Personnel

Reply:

Figure 3.13:



Cause:

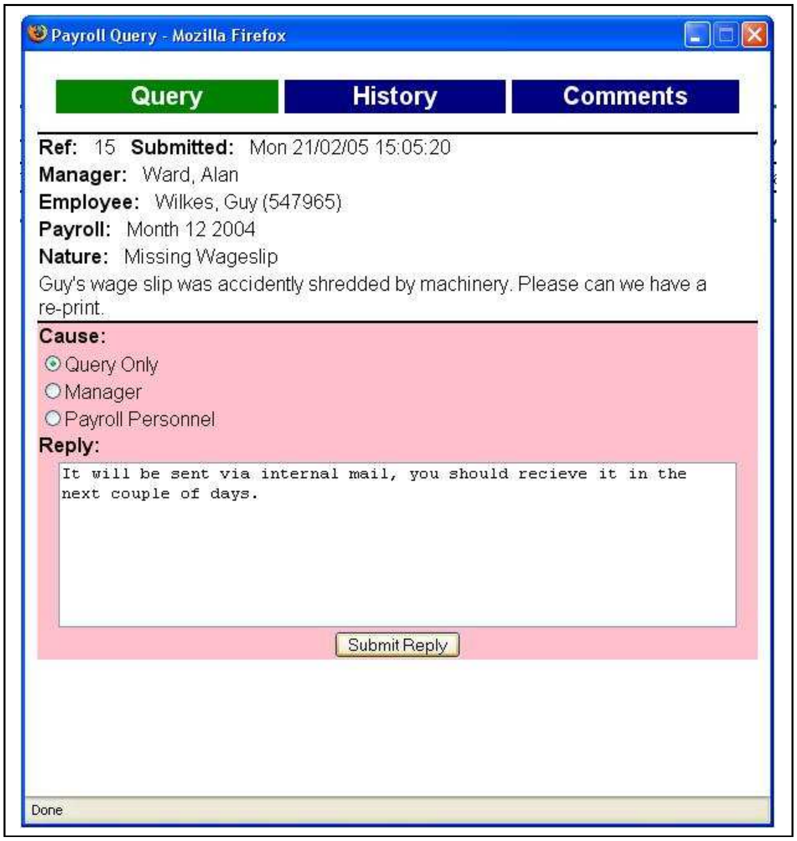
○ Query Only
○ Manager
● Payroll Personnel

Reply:

Robert Drake

Figure 3.14:

3. Type the reply to the query into the text box as shown in figure 3.15



Ref: 15 **Submitted:** Mon 21/02/05 15:05:20
Manager: Ward, Alan
Employee: Wilkes, Guy (547965)
Payroll: Month 12 2004
Nature: Missing Wageslip
Guy's wage slip was accidently shredded by machinery. Please can we have a re-print.

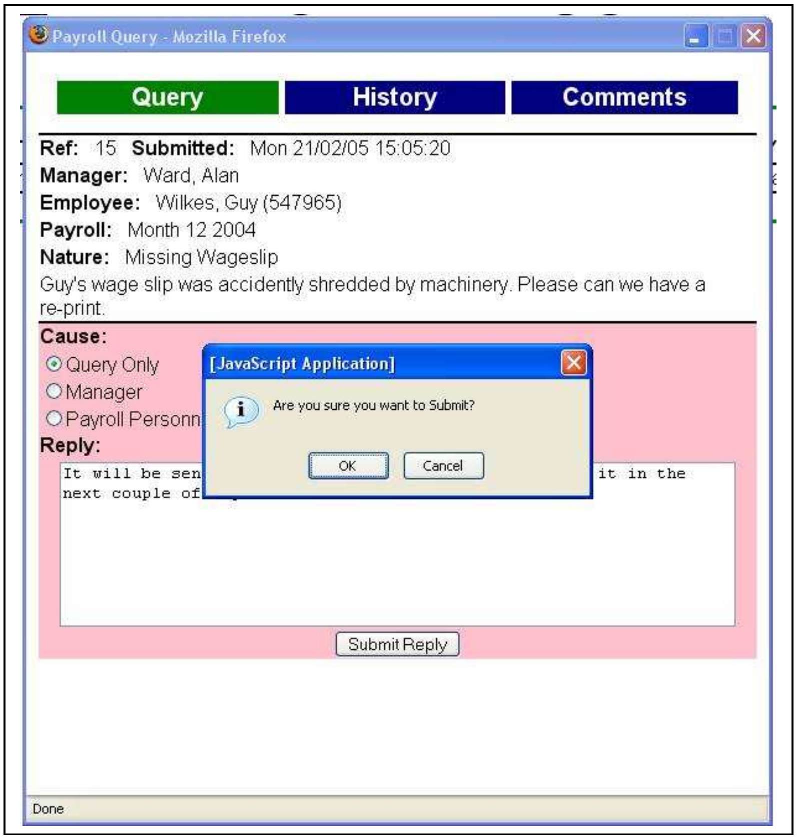
Cause:
● Query Only
○ Manager
○ Payroll Personnel

Reply:

```
It will be sent via internal mail, you should recieve it in the
next couple of days.
```

Figure 3.15:

4. You should see a dialogue box as shown in figure 3.16.
5. Left click on ”OK” if you are sure you want to submit or ”Cancel” if you don’t.



Ref: 15 **Submitted:** Mon 21/02/05 15:05:20
Manager: Ward, Alan
Employee: Wilkes, Guy (547965)
Payroll: Month 12 2004
Nature: Missing Wageslip

Guy's wage slip was accidently shredded by machinery. Please can we have a re-print.

Cause:
● Query Only
○ Manager
○ Payroll Personn

Reply:

It will be sen it in the next couple of

[JavaScript Application]

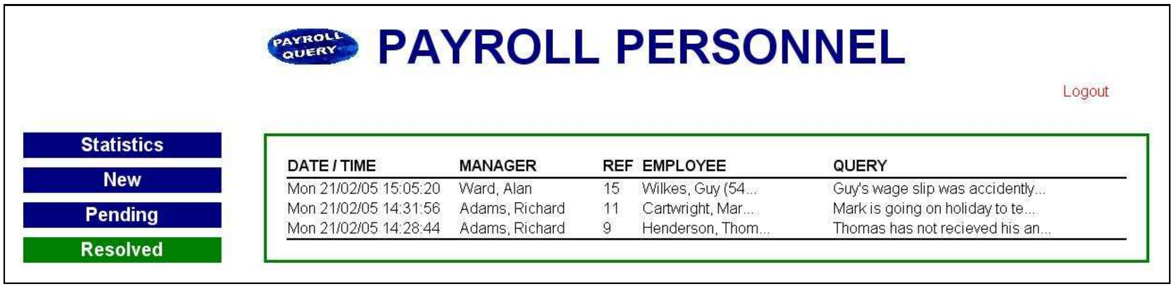
Are you sure you want to Submit?

Figure 3.16:

3.4 Resolved queries

Resolved queries are those that have been resolved by one or more of the Payroll Personnel.

Left click on ”Resolved” located towards the left side of the browser. The date/time of submission, the manager who submitted the query, the reference number assigned, employee’s details and the part of the query are displayed for the resolved queries shown in figure 3.17.



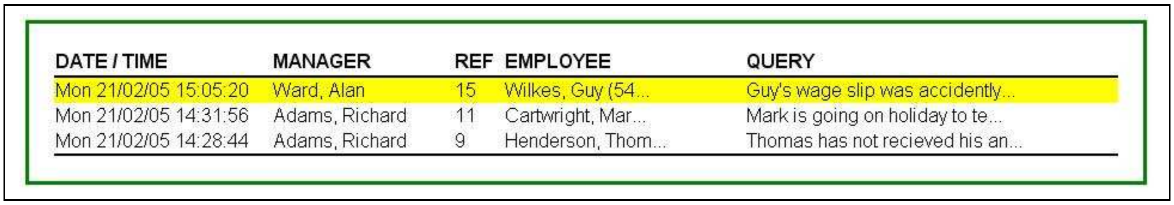
DATE / TIME	MANAGER	REF	EMPLOYEE	QUERY
Mon 21/02/05 15:05:20	Ward, Alan	15	Wilkes, Guy (54...	Guy's wage slip was accidently...
Mon 21/02/05 14:31:56	Adams, Richard	11	Cartwright, Mar...	Mark is going on holiday to te...
Mon 21/02/05 14:28:44	Adams, Richard	9	Henderson, Thom...	Thomas has not recieved his an...

Figure 3.17:

3.4.1 Viewing

To view a resolved query.

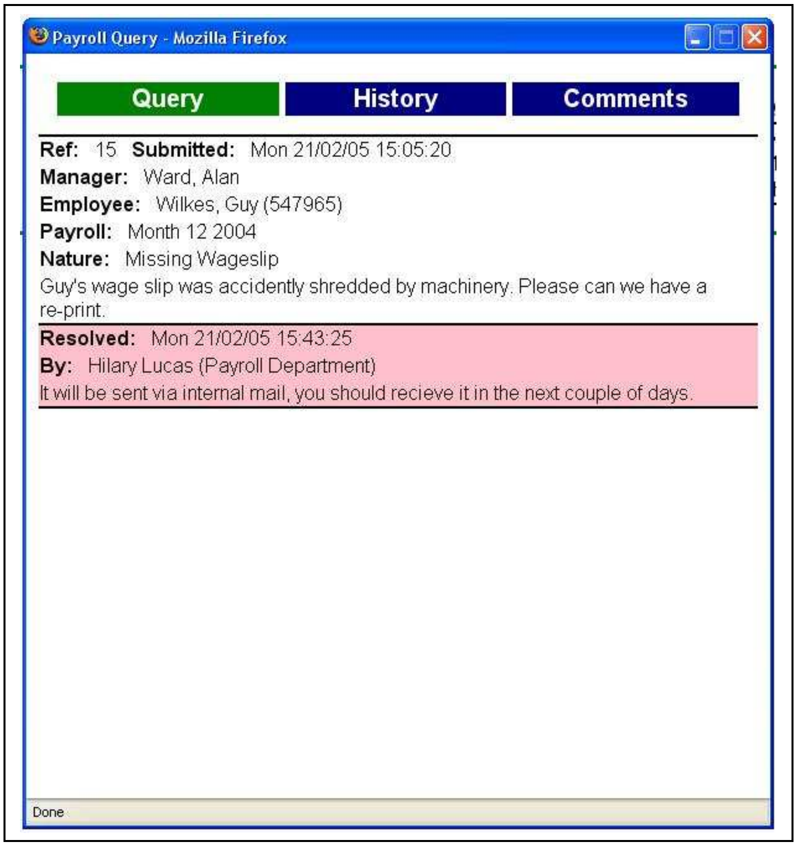
1. Move your mouse pointer over the query you wish to view.
2. You will notice that it is highlighted (shown in figure 3.18).
3. Left click on the highlighted query.



DATE / TIME	MANAGER	REF	EMPLOYEE	QUERY
Mon 21/02/05 15:05:20	Ward, Alan	15	Wilkes, Guy (54...	Guy's wage slip was accidently...
Mon 21/02/05 14:31:56	Adams, Richard	11	Cartwright, Mar...	Mark is going on holiday to te...
Mon 21/02/05 14:28:44	Adams, Richard	9	Henderson, Thom...	Thomas has not recieved his an...

Figure 3.18:

4. A new window should open like that in figure 3.19.
5. The window will contain the information submitted by the manager in the query along with the reply submitted by the payroll personnel.



Ref: 15 **Submitted:** Mon 21/02/05 15:05:20
Manager: Ward, Alan
Employee: Wilkes, Guy (547965)
Payroll: Month 12 2004
Nature: Missing Wageslip
Guy's wage slip was accidently shredded by machinery. Please can we have a re-print.

Resolved: Mon 21/02/05 15:43:25
By: Hilary Lucas (Payroll Department)
It will be sent via internal mail, you should recieve it in the next couple of days.

Figure 3.19:

3.4.2 Viewing history

To view the history of the query see section 3.2.2, this describes how to view the history of the query.

3.4.3 Viewing comments

To view the comments added to the query see section 3.2.3, this describes how to view the comments added to the query.

3.4.4 Adding comments

To add comments to the query see section 3.2.4, this describes how to add comments to the query.

Chapter 4

Payroll Manager

The payroll manager interface allows you to view statistics calculated from the payroll query database on a weekly basis. After logging in, the statistics shown will be for the current week. If you wish to view statistics for previous weeks see section 5.1, which describes how to change the week.

4.1 Changing the week

If you are compiling a report for a previous week it will be necessary to change the week for which the statistics relate.

To change the week.

1. Select the desired week from the drop down box as shown in figure 4.1.
2. Select the desired year from the drop down box to the right of the week.
3. Left click on ”Change Week”.

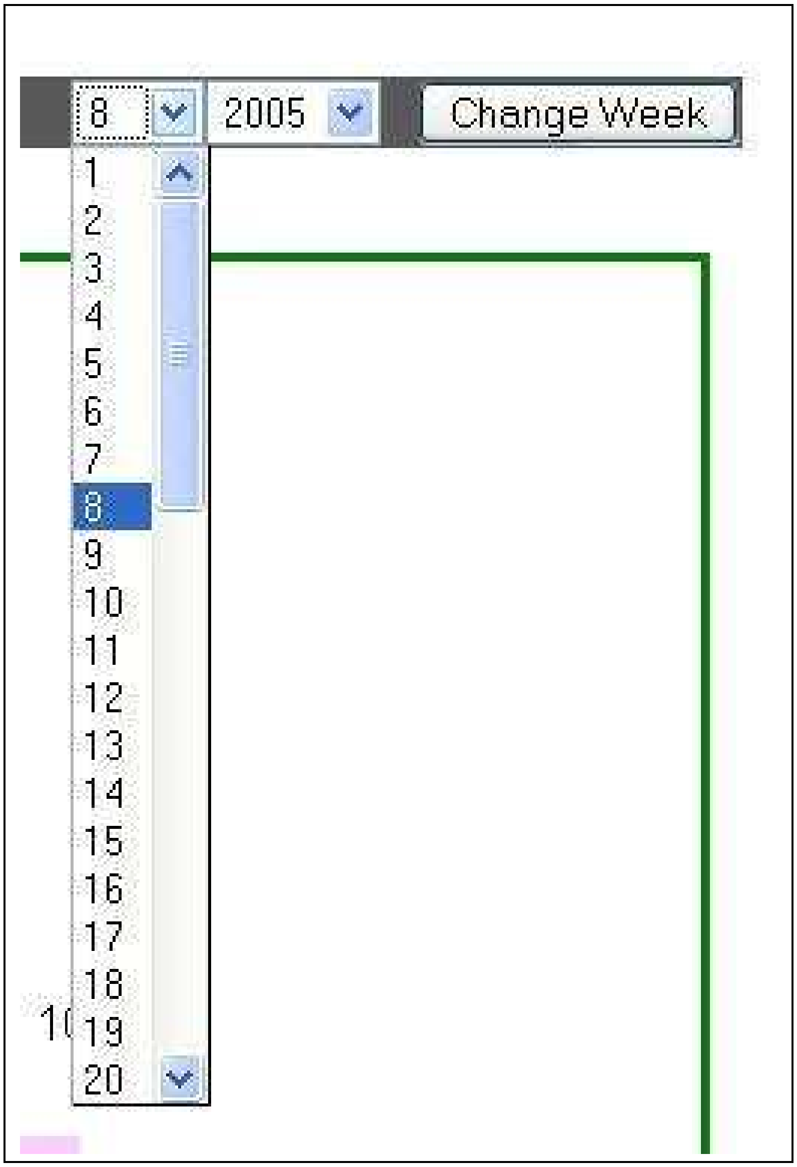
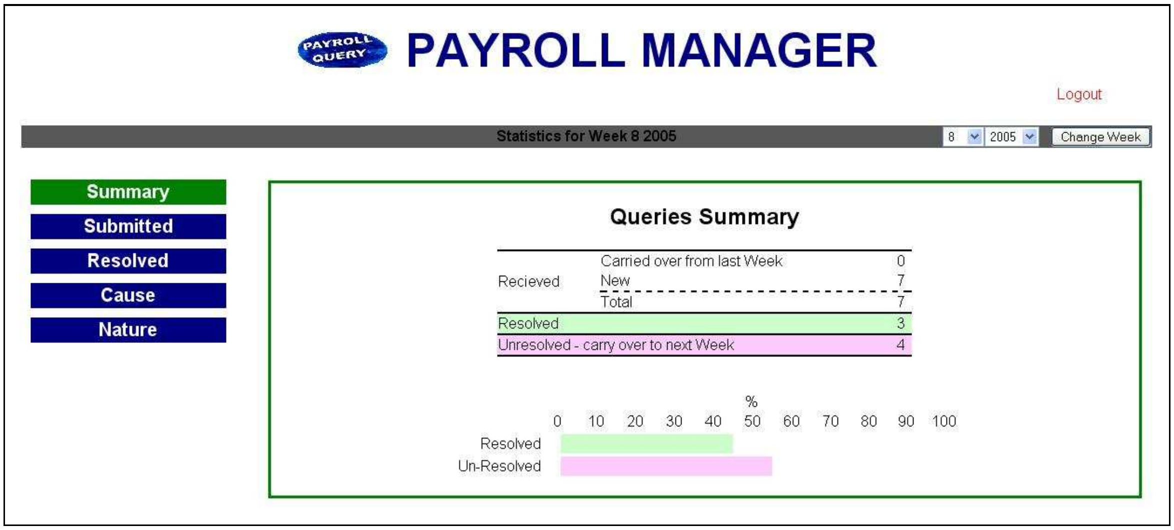


Figure 4.1:

4.2 Summary

Queries summary shows the amount of queries processed by the payroll department. The amount of queries received is broken down into unresolved queries carried over from the previous week and new queries received during the specified week. The bar chart shows the percentage of queries which were resolved and unresolved during that week.



Statistics for Week 8 2005

8 2005

Queries Summary

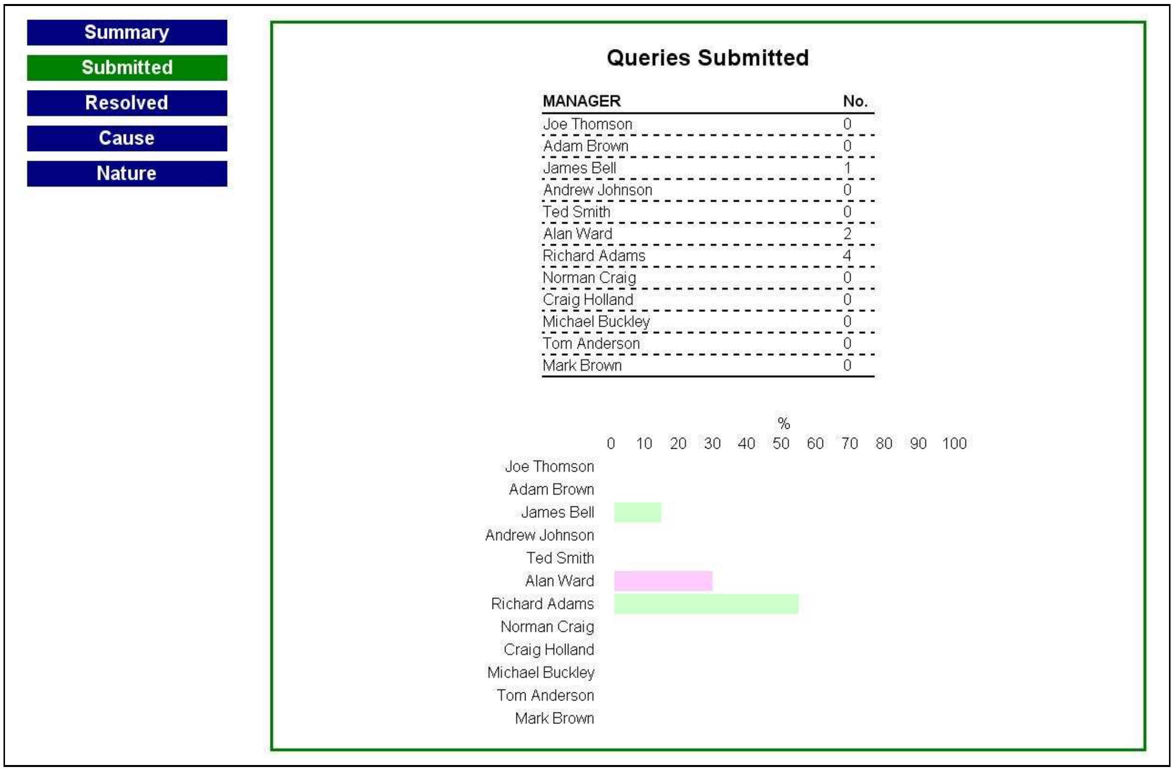
Recieved	Carried over from last Week	0
	New	7
	Total	7
Resolved		3
Unresolved - carry over to next Week		4

Figure 4.2:

Queries summary can be viewed by left clicking on ”Summary” located towards the left side of the browser.

4.3 Queries submitted

Queries Submitted shows the number of queries submitted by each manager during the specified week. The bar chart shows the percentage of total queries submitted by each manager during that week.



Queries Submitted

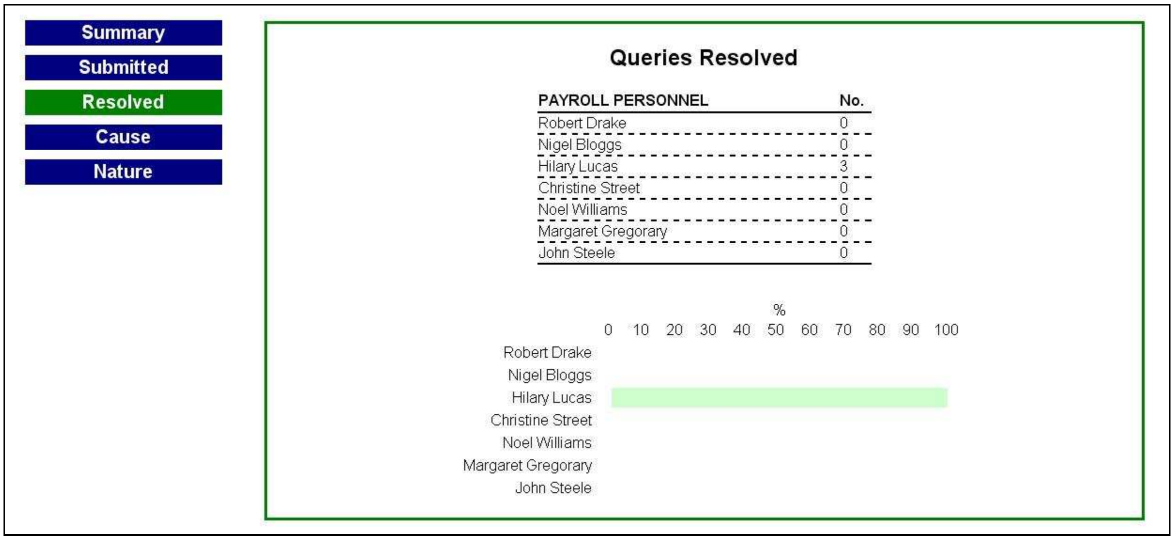
MANAGER	No.
Joe Thomson	0
Adam Brown	0
James Bell	1
Andrew Johnson	0
Ted Smith	0
Alan Ward	2
Richard Adams	4
Norman Craig	0
Craig Holland	0
Michael Buckley	0
Tom Anderson	0
Mark Brown	0

Figure 4.3:

Queries Submitted can be viewed by left clicking on ”Submitted” located towards the left side of the browser.

4.4 Queries resolved

Queries Resolved shows the number of queries resolved by each payroll personnel during the specified week. The bar chart shows the percentage of total queries resolved by each payroll personnel during that week.



Queries Resolved

PAYROLL PERSONNEL	No.
Robert Drake	0
Nigel Bloggs	0
Hilary Lucas	3
Christine Street	0
Noel Williams	0
Margaret Gregorary	0
John Steele	0

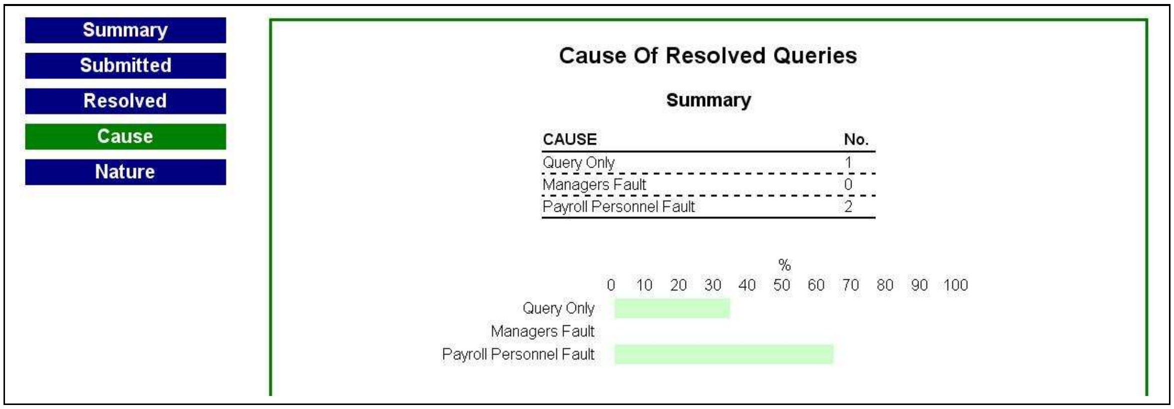
Figure 4.4:

Queries Resolved can be viewed by left clicking on ”Resolved” located towards the left side of the browser.

4.5 Cause of queries

Cause of Resolved Queries is composed of three sections.

The first, being a summary showing the cause of the queries that were resolved during the specified week.



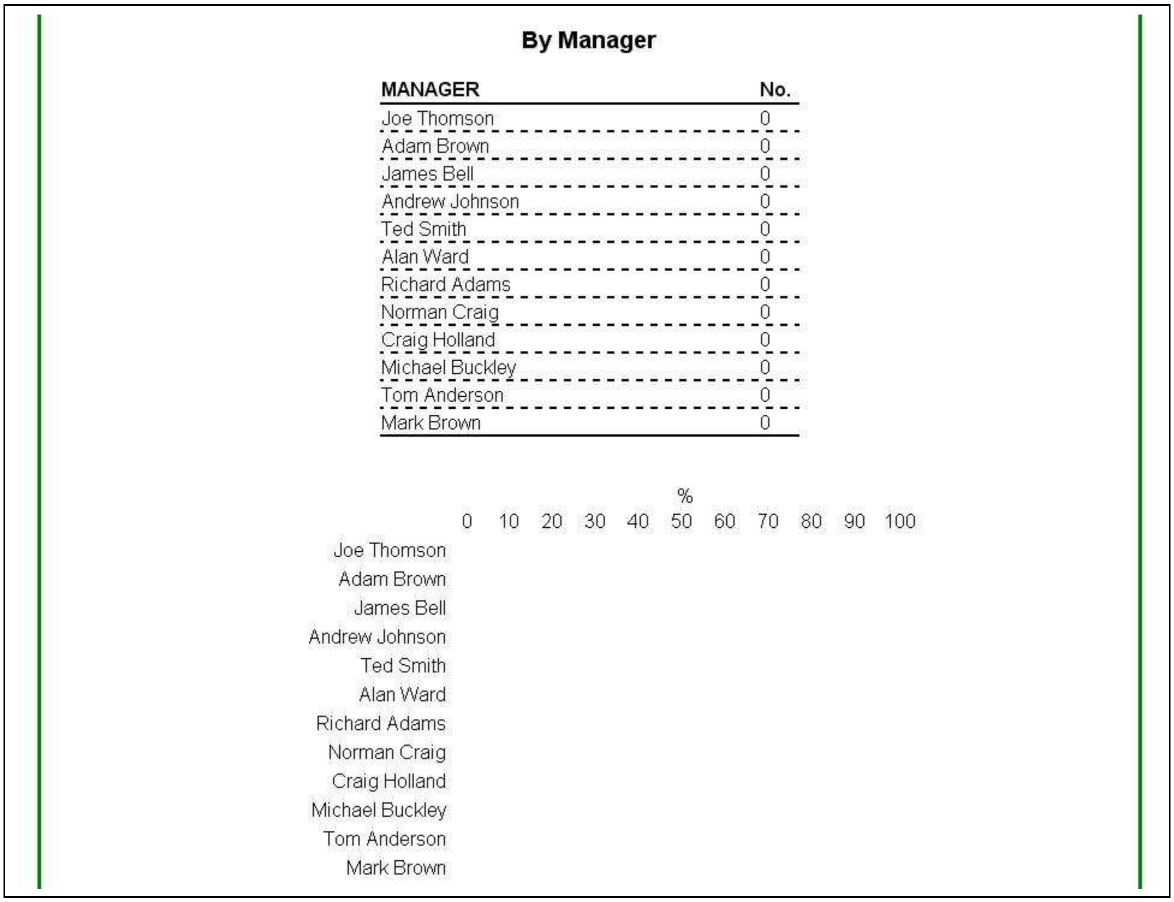
Cause Of Resolved Queries

Summary

CAUSE	No.
Query Only	1
Managers Fault	0
Payroll Personnel Fault	2

Figure 4.5:

The second, breaking down the queries that were caused by the Managers showing the number of queries caused by each manager for the specified week.

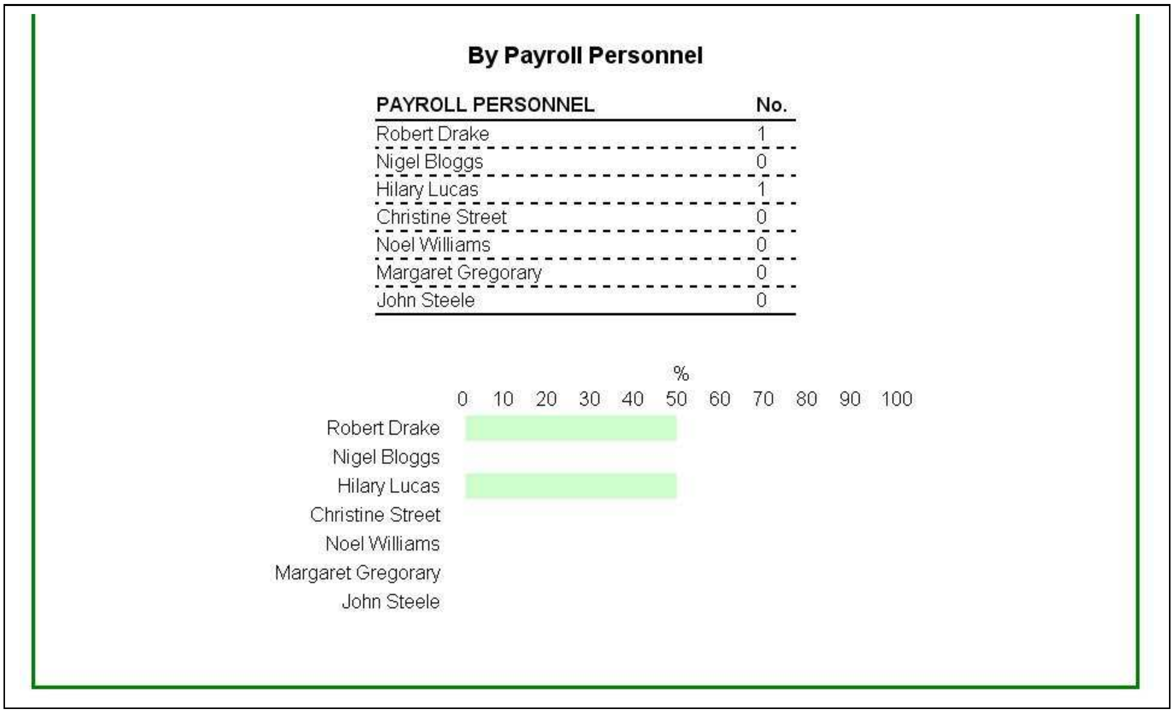


By Manager

MANAGER	No.
Joe Thomson	0
Adam Brown	0
James Bell	0
Andrew Johnson	0
Ted Smith	0
Alan Ward	0
Richard Adams	0
Norman Craig	0
Craig Holland	0
Michael Buckley	0
Tom Anderson	0
Mark Brown	0

Figure 4.6:

The third, breaking down the queries that were caused by the Payroll Personnel showing the number of queries caused by each Payroll Personnel for the specified week.



By Payroll Personnel

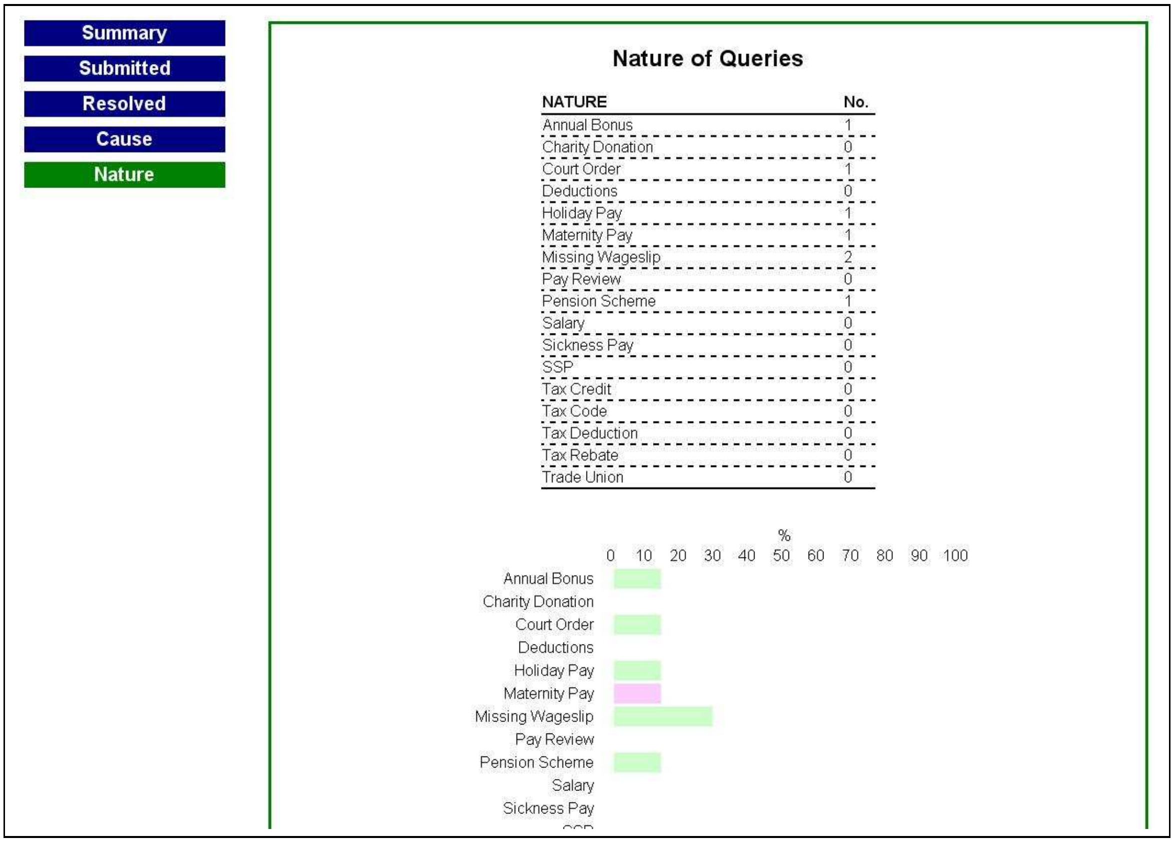
PAYROLL PERSONNEL	No.
Robert Drake	1
Nigel Bloggs	0
Hilary Lucas	1
Christine Street	0
Noel Williams	0
Margaret Gregorary	0
John Steele	0

Figure 4.7:

Cause of Resolved Queries can be viewed by left clicking on ”Cause” located towards the left side of the browser.

4.6 Nature of queries

Nature of queries shows the number of queries of each nature for the specified week. The bar chart shows the percentage of total queries of each nature during that week.



Nature of Queries

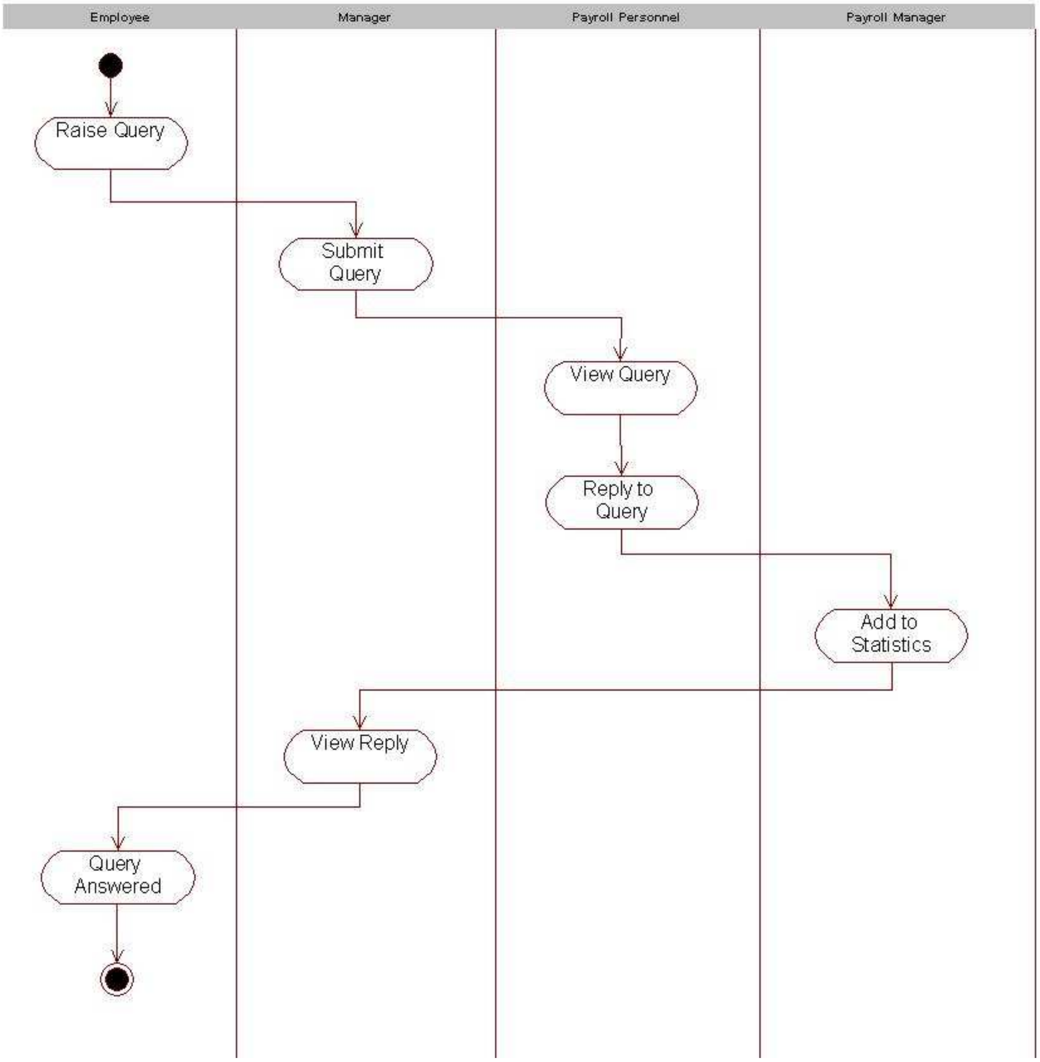
NATURE	No.
Annual Bonus	1
Charity Donation	0
Court Order	1
Deductions	0
Holiday Pay	1
Maternity Pay	1
Missing Wageslip	2
Pay Review	0
Pension Scheme	1
Salary	0
Sickness Pay	0
SSP	0
Tax Credit	0
Tax Code	0
Tax Deduction	0
Tax Rebate	0
Trade Union	0

Figure 4.8:

Nature of Queries can be viewed by left clicking on ”Nature” located towards the left side of the browser.

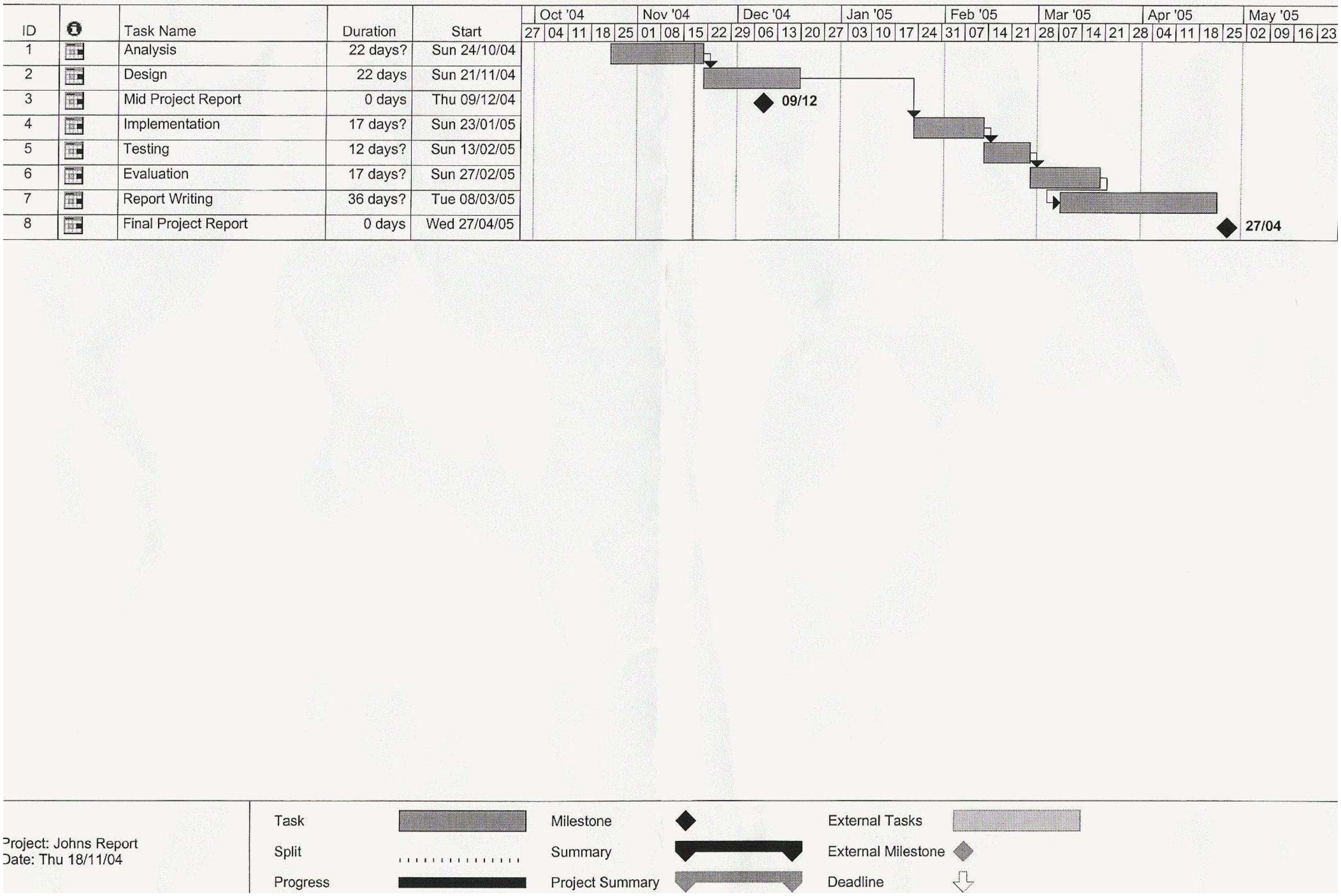
Appendix C

Payroll Query Process - Activity Diagram


Employee
Manager
Payroll Personnel
Payroll Manager
Raise Query
Submit Query
View Query
Reply to Query
Add to Statistics
View Reply
Query Answered

Appendix D

Gannt Chart



ID	🛈	Task Name	Duration	Start
1	▦	Analysis	22 days?	Sun 24/10/04
2	▦	Design	22 days	Sun 21/11/04
3	▦	Mid Project Report	0 days	Thu 09/12/04
4	▦	Implementation	17 days?	Sun 23/01/05
5	▦	Testing	12 days?	Sun 13/02/05
6	▦	Evaluation	17 days?	Sun 27/02/05
7	▦	Report Writing	36 days?	Tue 08/03/05
8	▦	Final Project Report	0 days	Wed 27/04/05

Project: Johns Report
Date: Thu 18/11/04

Appendix E

Interview Notes

*M*anager - Basharat Hussain

1. How long have you been in this role?

   About three years now.

2. How do you currently submit payroll queries?

   I usually send an email to the payroll department containing my query and await a reply. If it is really urgent I will ring or visit the payroll department.

3. How effective do you find this system?

   Reasonably effective.

4. What do you like about this system?

   Easy to use.

5. What do you not like about this system?

   - *It can often be a while before I receive a reply from the payroll department.*
   - *If I forget to include some details relating to the query I have to send them another email, delaying it being resolved.*
   - *It is hard to keep track of replies to previous queries should the employee re-query it.*

6. How would you improve the current system? What functionality would you like to see in a new system?

    - *Having a way of making sure all required details for the query are present before submitting the query.*
    - *Structured storage for all previously resolved queries for future reference.*

7. How many people carry out you role?
    About fifty on this site and probably about the same on the other three sites.

8. How many people carrying out your role would use the system simultaneously?
    Bearing in mind we have three separate shifts, so only a third of us will be on site at any one time. It would be a third if we were all submitting queries at once however it is unlikely, but possible after the payroll has just been run and the employees get their wage slip.

*P*ayroll Personnel - Hilary Rudd

1. How long have you been in this role?
    Over five years.

2. How do you currently submit payroll queries?

    - *After receiving the query (email) from the manager, we resolve the query and send an email reply back to the manager.*
    - *We complete paper query forms when completing more urgent queries over the telephone and window giving the reply verbally.*

3. How effective do you find this system?
    Not very effective.

4. What do you like about this system?
    Easy to use.

5. What do you not like about this system?

    - *No means of making sure the queries are resolved in the same order as they were received causing delays in some queries being resolved.*

- *Hard to track whether or not someone else is dealing with a particular query wasting payroll personnel time and again causing delays in some queries being resolved.*
- *No way for payroll personnel to add comments to queries.*
- *Two storage solutions: email folder and paper based forms make it hard to find previously resolved queries quickly.*
- *Managers can submit queries that don't contain the minimum details for us to resolve it wasting time replying asking for those details.*

6. How would you improve the current system? What functionality would you like to see in a new system?

   - *Having a way of making sure all required details for the query are present before the manager is allowed to submit the query.*
   - *Only one structured storage solution for all previously resolved queries for future reference (getting rid of paper based forms).*
   - *A History showing who has viewed and replied to each query.*
   - *A comments section for each query allowing payroll personnel to remind themselves and others what stage the query is at and how it was resolved in terms of payroll jargon.*

7. How many people carry out you role?

   There are seventeen payroll personnel in our office and we do the payroll for all sites.

8. How many people carrying out your role would use the system simultaneously?

   Although we have a variation in hours in which some of us work. At the peak time of the day (mid-day) a maximum of seven of us could be resolving queries at any one time.

*P*ayroll Manager - Jacqueline Buckley

1. How long have you been in this role?

   Over five years.

2. How do you currently submit payroll queries?

   - *I manually count all the queries (both email and paper forms) that are due to an error caused by either the managers or each of the payroll personnel to compile my error report.*

- *I keep my eye on the amount of queries throughout the day and allocate more payroll personnel to resolve queries if the numbers are high.*

3. How effective do you find this system?

   It works but drastically need improving to reduce the time spent on manual tasks.

4. What do you like about this system?

   Nothing really except it works and has been used for a number of years.

5. What do you not like about this system?

   - *There is no way of making sure the queries are resolved in the same order as they were received. If a query is not resolved within two days of receipt the department is breaking its service level agreement (SLA) to the company.*
   - *Hard to track which payroll personnel have completed which queries and produce reports containing information such as the number of queries resolved by each payroll personnel in order to measure their performance.*
   - *Two storage solutions: email folder and paper based forms make it hard to count queries quickly.*
   - *A lot of time is wasted manually counting queries for each person's error in order to compile the error report.*

6. How would you improve the current system? What functionality would you like to see in a new system?

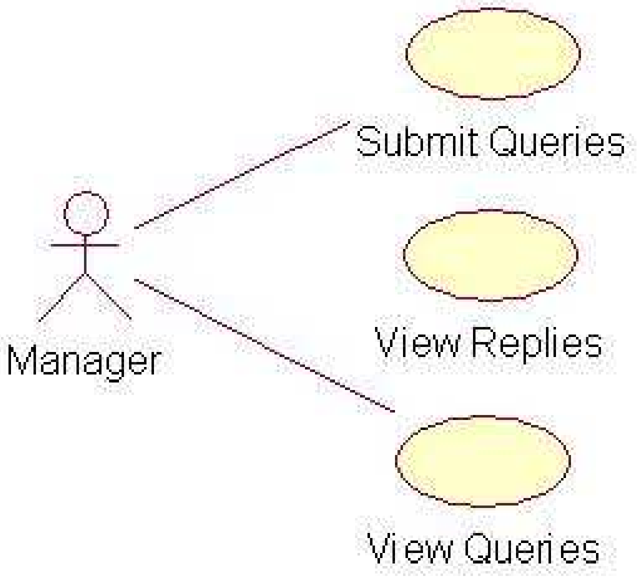
   - *A means of ensuring queries are resolved on a first come first served basis to ensure the department meets its SLA.*
   - *Only one structured storage solution for all previously resolved queries which will automatically count the queries matching report criteria.*
   - *A way of extracting information relating to the performance of each payroll personnel.*

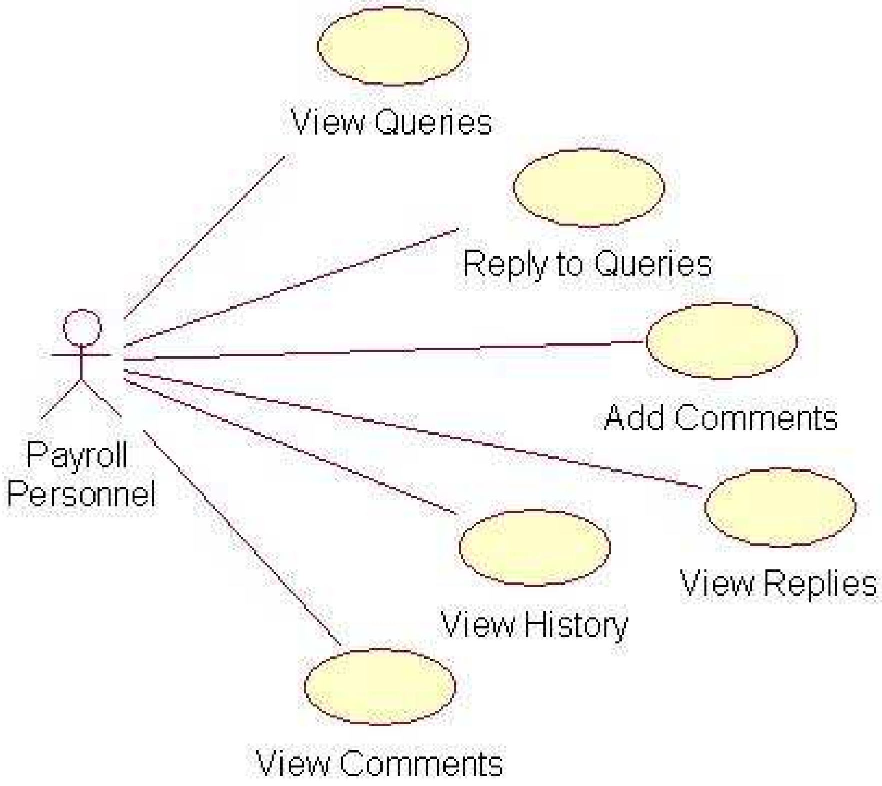
7. How many people carry out you role?

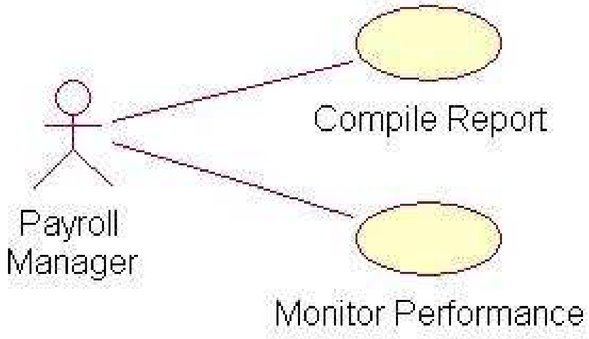
   There is just me.

Appendix F

Individual Use Case Diagrams




View Queries
Reply to Queries
Add Comments
View Replies
View History
View Comments
Payroll Personnel


Compile Report
Monitor Performance
Payroll Manager

Appendix G

Table Design - SQL statements

```
Create table if not exists manager(
  username CHAR(6) NOT NULL,
  password VARCHAR(10) NOT NULL,
  f_name VARCHAR(20) NOT NULL,
  l_name VARCHAR(20) NOT NULL,
  PRIMARY KEY(username)
);

Create table if not exists payroll_personnel(
  username CHAR(6) NOT NULL,
  password VARCHAR(10) NOT NULL,
  f_name VARCHAR(20) NOT NULL,
  l_name VARCHAR(20) NOT NULL,
  PRIMARY KEY(username)
);

Create table if not exists query(
  ref INT(5) NOT NULL auto_increment,
  f_name VARCHAR(20) NOT NULL,
  l_name VARCHAR(20) NOT NULL,
  payroll_no INT(6) NULL,
  query TEXT NOT NULL,
  date_time TIMESTAMP NOT NULL,
  manager_username CHAR(6) NOT NULL,
  PRIMARY KEY(ref),
  FOREIGN KEY(manager_username) references manager(username)
);

Create table if not exists reply(
  date_time TIMESTAMP NOT NULL,
  query_ref INT(5) NOT NULL,
  payroll_personnel_username CHAR(6) NOT NULL,
  reply TEXT NOT NULL,
  PRIMARY KEY(date_time, query_ref, payroll_personnel_username),
  FOREIGN KEY(query_ref) references query(ref),
  FOREIGN KEY(payroll_personnel_username) references payroll_personnel(username)
);

Create table if not exists comment(
  date_time TIMESTAMP NOT NULL,
  query_ref INT(5) NOT NULL,
  payroll_personnel_username CHAR(6) NOT NULL,
  comment TEXT NOT NULL,
  PRIMARY KEY(date_time, query_ref, payroll_personnel_username),
  FOREIGN KEY(query_ref) references query(ref),
  FOREIGN KEY(payroll_personnel_username) references payroll_personnel(username)
);
```

```
Create table if not exists view(
  date_time TIMESTAMP NOT NULL,
  query_ref INT(5) NOT NULL,
  payroll_personnel_username CHAR(6) NOT NULL,
  PRIMARY KEY(date_time, query_ref, payroll_personnel_username),
  FOREIGN KEY(query_ref) references query(ref),
  FOREIGN KEY(payroll_personnel_username) references payroll_personnel(username)
);
```

Appendix H

Usability Evaluation Questionnaire

The usability evaluation questionnaire was completed by eight experts who are as follows: Thomas Anderson **TA**, Richard Bevan **RB**, Andrew Durrans **AD**, Jacqueline Buckley **JB**, Mark Brown **MB**, Basharat Hussain **BH**, Mohammed Mustafa **MM** and James Alexander **JA**.

Learnability

1. How easy did you find the system was to learn?

   TA Very easy to learn, simple uncluttered layout, clear description of buttons and forms.

   RB The system is very easy to follow and information is easily found.

   AD Simple to use, menu screens clear.

   JB Extremely easy.

   MB The minimalist design made for ease of navigability. Prior experience in related payroll software was transferable to reduce the overall learning curve.

   BH Very easy.

   MM Easy.

   JA Very easy.

2. Did you feel the need to refer to the Instruction Manual? If yes, how often and did it resolve any problems you were experiencing?

   TA Yes, I consulted the instruction manual at the start. I did not need to consult it again.

RB I felt that the system allowed me to navigate without using the instruction manual.

AD No, fairly self explanatory.

JB Never needed to refer to manual.

MB The instruction manual provided a useful guide to clarifying some of the features which were not clear e.g. the interaction between when and how the query state changes from pending to resolved.

BH No, didn't need it. It was already explained in much detail.

MM No. Very well explained.

JA No, I did not need it.

3. Did you find the navigational structure and page layout was easy to follow?

   TA Yes, all the functionality was availible in view. Buttons and links were consistently placed.

   RB Yes, with ease.

   AD Yes.

   JB Yes.

   MB A single layer vertical navigation bar made for effective browsing, further supporting the minimalist approach. The form was laid out in a familiar manner i.e. appropriate use of different form components such as combo boxes and radio buttons.

   BH Yes, the bars on the left of the screen eg ”Summary” or ”Submitted” etc made navigation very easy and were in bold.

   MM Yes.

   JA Yes.

Efficiency

4. Did you experience any noticeable delays when using the system?

   TA No, the system was very quick. No delays at all.

   RB Minor restrictions, one to connections. No system hangs or crashes.

   AD No, the system was very efficient.

   JB No.

   MB No delays experienced in any tasks carried out thus far.

   BH No, it was very fast on the computer used.

   MM No, was quick.

JA It was very fast with no delays.

5. Did you find the page layout supported the order of completing tasks?

   TA Yes.

   RB There was a linear flow to links which allowed tasks to be completed in order.

   AD Yes, definitely.

   JB It left nothing to chance, all details asked for were relevant.

   MB The order of the links on the navigation bar, did follow a logical sequence of the order in which I would have expected tasks to be performed.

   BH Yes, the data in tabulated form was clear and easy to follow with more details available if needed after the quick search.

   MM Yes, easy to follow.

   JA Yes.

Memorability

6. Did you find the system easy to remember how to use?

   TA Yes, because all the functionality is placed in view and consistently between pages.

   RB The system is simplistic and easy to remember.

   AD Yes, its clarity allowed this.

   JB Yes, only took a few minutes.

   MB Keeping the front-end uncluttered and using an intuitive template, made the website very simple to learn and recall on subsequent visits.

   BH Yes, no problems.

   MM Yes.

   JA Yes, it was very logical.

7. Did you find the navigational structure and page layout supported this?

   TA Yes.

   RB Yes I did.

   AD Yes.

   JB Yes.

   MB Setting links out in a manner by which a user expects to perform tasks, assists in being able to form spatial (cognitive) representations. In knowing the normal order of the tasks, I was soon able to subconsciously recall the layout of the front-end.

BH Page layout was easy to use by a relatively clear screen with easily recognisable and usable menu style.

MM Very clear, good page layout.

JA Yes, the navigational structure made the system easy to use.

Errors

8. Did any errors occur whilst using the system? What was the nature of those errors?

   TA One error (mine) which was I forgot to fill in the month field in the query form.

   RB Yes, incomplete fields.

   AD No.

   JB No errors.

   MB The form for entering new queries seems to have no validation on the ’First Name’ and ’Last Name’ fields. This could be a problem if for example a payroll number is accidentally copied from an external source and pasted into a wrong field e.g. the name fields.

   BH No.

   MM No.

   JA No errors.

9. Were the error messages friendly and did they help you resolve the problem?

   TA Yes, the field that I forgot to fill in was highlighted and a message box appeared inform me.

   RB Yes, they highlighted where the error was.

   AD NA.

   JB Yes, made deliberate mistake to see.

   MB The error message belonging to the form for entering two queries, does not give an idea as to the nature of the problem. Although the fields with which either no data or wrongly formatted data has been entered are visibly highlighted, no clue is given in the error message as to what is wrong for each identified field. The user must guess through a trial and error approach.

   BH NA.

   MM NA.

   JA See above.

Satisfaction

10. Did you enjoy using the system?

    TA Yes.

    RB Yes.

    AD Yes, quite cheerful.

    JB Yes, please implement in Littlewoods.

    MB Apart from the minor problem with the validation, the system was fuss free and its simplicity made for an amiable experience.

    BH Yes.

    MM Yes.

    JA Yes.

11. How did you find the colours and aesthetics of the system?

    TA Good, apart from the yellow on the menu mouse over - maybe something that provided a better contrast with the lettering.

    RB The colour was easy on the eyes.

    AD Appropriate, no focus distracted off the menus.

    JB Pleasing and easy to read.

    MB The overall colour scheme using mainly primary colours helps in making clear distinctions between areas of a web page. A slight concern should be noted that the use of yellow for rollover links is questionable given both the background colour of the system and the writing within the links are white. I believe the closeness in luminance could be a problem for the visually impaired!

    BH Very bright, modern and clear.

    MM Pleasing on the eye.

    JA Yes, the colours helped to give the system a good image.

Control

12. Were there any occasions when you felt you were not in control of the system? If yes, what happened on these occasions?

    TA No.

    RB Yes, the system moved new queries to pending when they had been viewed, without asking the user.

    AD No.

JB Password system does not provide feedback on errors.

MB No, I did feel in control in all tasks performed using this system.

BH No.

MM No.

JA No.

Skills

13. How professional did you find the system?

    TA Very professional - no problems using the system.

    RB The construction of the system seemed professional.

    AD Very, appeared professional.

    JB Excellent.

    MB The layout has the standardised look and feel of both other payroll query software and also of email clients. The layout, colours and fonts chosen, all add to the professionalism of the system.

    BH Very, modern and accessible.

    MM Very good.

    JA It was more professional than most systems I use at work.

14. Did you find it enhanced your skills?

    TA NA.

    RB NA.

    AD Yes, showed me new methods.

    JB Yes, it would in time.

    MB Although I don't believe I could honestly say my skills have been enhanced, I feel the system has helped increase my productivity.

    BH I didn't need any computer skills to use it.

    MM Yes.

    JA Would need to use it more to enhance my skills.

15. Did you feel valued whilst using the system?

    TA The personalised greeting make it feel that the system is providing a service to you.

    RB Yes, everything was personalised by using system logins.

    AD Yes, seemed to appreciate the role played.

JB Yes, because it is user friendly.

MB Receiving responses from HR staff in a timely fashion, I did feel valued whilst using the system.

BH Yes.

MM Yes.

JA Yes.

Privacy

16. How confident did you feel that the system complied with the data protection legislation practiced in your role?

TA This would depend on where the system was installed. If it was installed within a secure internal network I would be happy.

RB The system conforms to the DPA by protecting user informationa and not passing it on to third parties.

AD Fairly, system seemed secure to the user with password system etc.

JB Very confident, information given is covered by the DPA.

MB To my knowledge, the system did seem to conform to the DPA legislation whilst performing various tasks. Notably, only storing disputes for a certain period after being resolved ensures data is not kept for longer than is necessary. The details in the form are appropriate to the nature for which they will be processed. On the proviso of improved validation, query records should also be fairly accurate.

BH Unkown.

MM As far as I know.

JA Fully.